

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA APLIKOVANÉ INFORMATIKY

Aplikovaná informatika
Applied Informatics

Student: Tomáš Čeladník
Vedoucí diplomové práce: Ing. Pochyla Martin, Ph.D.

Ostrava 2011

Zadání diplomové práce

Student: Tomáš Čeladník

Studijní program: Systémové inženýrství a informatika

Studijní obor: Aplikovaná informatika

Téma: Analýza a srovnání robustních internetových aplikací

Analysis and Comparison of Rich Internet Applications Development

1. Úvod

2. Vývoj aplikací v prostředí internetu

3. Trendy vývoje webových aplikací

4. Srovnání vybraných RIA technologií

5. Návrh a realizace demonstrační aplikace

6. Závěr

Seznam použité literatury

DEWSBURY, R. Google Web Toolkit Applications, Prentice Hall, 2007. 574 s. ISBN 978-0-321-50196-7

TRETOLA, R.; BARBER, S.; ERICKSON, R. *Professional Adobe Flex 2*. John Wiley and Sons, 2007. 687 s. ISBN 0470102675.

ZAKAS, N.; McPeak, J.; Fawcett, J. *Professional Ajax*. Wiley Publishing, Inc., 2006. 406 s. ISBN 978-0-471-77778-6

Seznam zkratk

Prohlášení o využití výsledků diplomové práce

Přílohy

Seznam doporučené odborné literatury:

Vedoucí diplomové práce: Ing. Pochyla Martin, Ph.D.

Datum zadání: 26. listopadu 2010

Datum odevzdání: 29. dubna 2011

Prohlášení

Prohlašuji, že předložená diplomová práce je mým původním autorským dílem, které jsem vypracoval samostatně za použití literatury a podkladů, které jsou v seznamu uvedeny.

Děkuji vedoucímu práce Ing. Pochyla Martin, Ph.D., za odbornou pomoc při tvorbě diplomové práce.

Ostrava

29. 4. 2011

.....

podpis

Obsah

1. Úvod	1
2. Vývoj a součásti aplikací v prostředí internetu	2
2.1 Historie „World Wide Web“	3
2.2 Vývoj webových aplikací	4
2.3 Druhy aplikací	12
2.4 Webové aplikace	13
2.4.1 Klient - server	13
2.4.2 Peer-to-peer	14
3. Trendy vývoje robustních internetových aplikací	15
3.1 Zkratka RIA	15
3.2 Definice pojmu RIA	15
3.3 Počátky Rich Internet Applications	16
3.3.1 Remote scripting – vzdálené skriptování	16
3.3.2 X-Internet	18
3.4 Oblasti působnosti RIA aplikací	19
3.5 Rich Internet Application	20
3.6 Vlastnosti RIA aplikací	22
3.7 Výhody a nevýhody RIA aplikací	23
3.8. Současné RIA technologie	24
3.8.1 AJAX versus zbytek světa	24
3.8.2 Zbytek světa	25
3.9 Shrnutí	25
4. Srovnání vybraných RIA technologií	26
4.0.1 Zvolený výběr RIA technologií	26
4.0.2 Analýza RIA technologií	27
4.0.3 Metodika bodového ohodnocení jednotlivých technologií	28
4.0.4 Stanovení důležitosti kritérií	28
4.1 Adobe Flex	29
4.1.1 Historie	30
4.1.2 Deklarativní jazyk pro popis uživatelského rozhraní - MXML	33
4.1.3 ActionScript	34
4.1.4 Vývojové prostředí – Adobe Flash Builder	35
4.1.5 Runtime prostředí	36

4.1.6 Kritéria hodnocení	37
4.2 Microsoft Silverlight	39
4.2.1 Historie	40
4.2.2 Značkovací jazyk - eXtensible Application Markup Language	43
4.2.3 C# a Visual Basic .NET.....	44
4.2.4 Vývojové prostředí – Expression Blend 4, Visual Studio 2010, Visual Web Developer 2010	45
4.2.5 Runtime prostředí – zásuvný modul Silverlight plugin.....	46
4.2.6 Kritéria hodnocení	47
4.3 JavaFX.....	50
4.3.1 Historie	51
4.3.2 Skriptovací jazyk - JavaFX Script.....	52
4.3.3 Vývojové prostředí – NetBeans a Eclipse	53
4.3.4 Runtime prostředí	55
4.3.5 Kritéria hodnocení	55
4.4 Google Web Toolkit.....	57
4.4.1 Historie	58
4.4.2 Programovací jazyk Java	59
4.4.3 Komponenty pro tvorbu uživatelského rozhraní	59
4.4.4 Vývojové prostředí	61
4.4.5 Kritéria hodnocení	61
4.5 OpenLaszlo.....	63
4.5.1 Historie	64
4.5.2 Značkovací jazyk LZX	64
4.5.3 Vývojové prostředí	65
4.5.4 Hodnocení kritérií.....	65
4.6 Matice prostých a vážených užitečností	67
5. Návrh a realizace demonstrační aplikace	68
5.1 Návrh aplikace - „Nenech bílou vyhrát“	69
5.2 Realizace aplikace - „Nenech bílou vyhrát“	70
5.3 Výsledná podoba aplikace	73
6. Závěr.....	75
Seznam použité literatury	78
Knihy	78

Elektronické publikace	78
Seznam zkratk.....	81
Přílohy	85

1. Úvod

„Internet“ od jeho vzniku prošel značným vývojem a nabízí stále nové možnosti. Po dlouhou dobu představovaly nejpoužívanější formu prohlížení internetu klasické webové stránky. Moderní doba, časový vývoj a neustále rostoucí požadavky uživatelů zapříčinily, že tato varianta řešení přestává být efektivní a pohodlná. Bylo nutné vytvořit internetovou aplikaci, která přináší uživateli rychlejší, přehlednější a efektivnější ovládání, usnadňuje práci a v některých případech by nahradila programy nainstalované na počítači. V dnešní době existují webové aplikace, které plní funkce podobné textovým editorům, grafickým programům, prezentačním programům apod. Ať již vědomě či nevědomě se s těmito webovými aplikacemi setkal určitě skoro každý uživatel internetu. Internetové aplikace se staly významným fenoménem světa IT a prošly značným vývojem v posledních letech. Vývoj aplikací v internetu není jednoduchá záležitost. Programátoři a vývojáři se často během vývoje potýkají s řadou úskalí, např. s nekompatibilitou jejich aplikace v různých internetových prohlížečích, s obtížným výběrem vhodných technologií pro jejich tvorbu na straně klienta, tak na straně serveru). Moderní technologie poskytují vývojářům jednoduché řešení v podobě komplexního balíku vývojových nástrojů, technologií a postupů pro snadné vytvoření webových aplikací.

Aplikace, které jsou vytvořeny pomocí těchto technologií, obecně nazýváme „RIA“. Daný pojem už není tolik neznámý, jelikož se často objevuje v mnoha odborných článcích, které se zabývají současnými trendy vývoje webových aplikací či internetu. Odborníci nazývají „RIA“ jako další generaci webových aplikací. Název „RIA“ se poprvé použil v devadesátých letech dvacátého století společností Macromedia, která od roku 2005 vlastní společnost Adobe Systems. Největší rozvoj nastal až s postupným vývojem webových standardů (např. AJAX, HTML 5) a odstraňováním rozdílů mezi internetovými prohlížeči. K nárůstu počtu aplikací přispělo i masové rozšíření technologií Java a Flash do webových prohlížečů, jelikož většina nově vytvořených interaktivních aplikací už nepotřebovala pro svůj běh instalovat další rozšíření. I když vývoj „RIA aplikací“ probíhá již několik let, domnívám se, že je stále ve svém počátku a pokud bude pokračovat, mohou nám nabídnout ještě řadu zajímavých vlastností, prvků a možností.

Práce si klade za cíl zmapovat současné technologie pro tvorbu robustních internetových aplikací (Rich Internet Applications), pokusit se vymezit a definovat oblasti působnosti RIA aplikace. Dalším cílem práce je samotná analýza RIA technologií, ve které

jsou detailně srovnány možnosti vybraných technologií a výsledné ohodnocení ukazující jejich vlastnosti v porovnání s ostatními. Výsledkem další části práce je návrh a realizace demonstrační aplikace, která ilustruje možnosti vývoje aplikací pomocí zvolené technologie pro tvorbu současných bohatých internetových aplikací - RIA.

2. Vývoj a součásti aplikací v prostředí internetu

O správné definici slova „internet“ s malým počátečním písmenem a slova „Internet“ s velkým písmenem se vedou značné rozpory. Obě verze mají své vášnivé stoupence. Jedni tvrdí, že Internet je jméno unikátní světové sítě, a tudíž je velké písmeno na začátku v pořádku (tak jako se píše Miloš, a ne miloš). Druzí ovšem oponují tím, že internet je obyčejné slovo popisující počítačovou infrastrukturu a že o žádné jméno nejde, pro velké písmeno tedy neexistuje žádný rozumný důvod (tak jako píšeme rádio, televize či telefon, a nikoliv Rádio, Televize nebo Telefon). V čem je tedy mezi nimi rozdíl? Pojem „Internet“ se používá od dob jeho vzniku a sděluje vlastní název sítí propojených do samostatného velkého celku. Kdežto pod pojmem slova „internet“ s malým počátečním písmenem si můžeme představit pouze technologickou infrastrukturu domácích či firemních prvků do jedné menší sítě. Poté lze „Internet“ pochopit jako celosvětové sloučení těchto malých sítí do jedné společné. [14] Proč vlastně poukazuji na tuto skutečnost? Nejsem zastáncem ani jedné z verzí a proto předem upozorňuji čtenáře (stoupence jedné z verzí), že v rámci své diplomové práce se nezaobírám tím, zda psát velké či malé počáteční písmeno ve slově „internet“. Snažím se dodržovat pravidla českého pravopisu, proto budu používat velké písmeno pouze na začátku věty.

V roce 1945, v červencovém čísle amerického časopisu „The Atlantic Monthly“ publikoval Vannevar Bush (1890-1974) svůj světoznámý „As We May Think“, jenž bývá považován za jeden ze základních kamenů informační vědy. Tento článek, který se týkal využití počítačů pro komunikaci je zajímavý tím, že byl napsán mnoho let před tím, než byly pro tuto úlohu skutečně poprvé počítače použity. První testovací síť byla instalována počátkem roku 1968 v Národní výzkumné laboratoři ve Velké Británii. Tato síť však neopustila hranice jedné budovy. Jak už to bývá, přišel požadavek na vybudování podobné sítě a zároveň i potřebné finanční prostředky z resortu obrany, konkrétně od grantové agentury ministerstva obrany USA s názvem ARPA. Podle této grantové agentury byla experimentální síť, která vznikla v roce 1969, také pojmenována jako ARPANET. V roce 1980 přišel ve švýcarském institutu pro jaderný výzkum CERN Tim Berners-Lee s myšlenkou

hypertextu, což mělo usnadnit sdílení a aktualizaci informací mezi výzkumníky. V roce 1989 již měl CERN největší internetový server v Evropě a Tim Berners-Lee oživil tehdy zapomenutou myšlenku hypertextu. Rokem 1993 začal Internet v USA prožívat nebyvalý rozmach, k Internetu byl připojen Bílý dům (White House, Washington, D. C). Byl vyvinut standard WWW, existuje již 50 WWW serverů. Od roku 1993 do roku 1995 se zdvojnásobil počet připojených počítačů k Internetu. V roce 1995 je celkem v USA k Internetu připojeno na dva milióny počítačů. Na celém světě je odhadováno v roce 1995 na 20 miliónů uživatelů Internetu, v roce 2000 již pak přes 300 miliónů. [15]

2.1 Historie „World Wide Web“

Projekt WWW byl zahájen v roce 1989 ve výzkumných laboratořích CERN v Ženevě ve Švýcarsku. Cílem projektu bylo navrhnout mechanismus sdílení a zveřejňování informací týkajících se řešených výzkumných projektů. První prototyp prohlížeče a editoru dokumentů vyvinul Tim Berners-Lee v prostředí systému NextStep. První veřejně dostupná verze prohlížeče podporovala pouze řádkový režim prohlížení dokumentů a byla uvolněna v roce 1992 (Line Mode). K prohlížení stránek sloužil prezentační systém GOPHER. Počátkem roku 1993 již pracovalo asi 50 serverů po celém světě. V polovině roku byl dokončen první grafický prohlížeč dokumentů - NCSA Mosaic pro prostředí X-Window, jehož autorem byl Marc Andreessen pracující v NCSA. Ve srovnání s prohlížečem v prostředí NextStep byl sice funkčně omezenější, ale podporoval barevné obrázky a byl volně šířitelný. Od té doby začíná prudce růst počet instalací prohlížečů a serverů www. V březnu 1994 byla uspořádána první mezinárodní konference věnovaná pouze systému WWW ve středisku CERN. Koncem roku 1994 přestává být CERN centrem vývoje world wide web a předává vývoj do francouzského institutu INRIA (Institut national de recherche en informatique et en automatique). V září 1994 byla založena nekomerční organizace World Wide Web Consortium, která má za cíl další vývoj a standardizaci systému world wide web. Je financována komerčními firmami, ale poskytuje výsledky svého vývoje zdarma veřejnosti. Hlavní týmy koordinátorů jsou soustředěny v MIT LCS (Laboratory of Computer Science) a INRIA. [15] Počátkem roku 1995 začíná obrovský boom systému hypertextových odkazů, své servery si instalují přední počítačové firmy a další komerční organizace, objevují se nové prohlížeče a první knihy o systému propojených hypertextových odkazů. Informace o systému WWW se začínají objevovat v populárních časopisech a motivují uživatele počítačů k připojení na internet. Snadné ovládání prohlížečů přitahuje uživatele, kteří dříve nenacházeli vhodné nástroje pro hledání a zobrazování informací v počítačových sítích. Množství dostupných informací roste

závratným tempem. Úspěch world wide webu také do jisté míry podnítil rozvoj firem poskytujících komerční připojení na síť internet a projekty zvýšení přenosových kapacit počítačových sítí. WWW byl založen zejména na následujících konceptech:

- URI neboli URL - řetězec, který pojmenovává a identifikuje dostupný zdroj
- HTTP - protokol pro přenos webových stránek, který je založen na protokolu FTP
- HTML - formát pro zápis stylovaných a strukturovaných
- textových dokumentů.

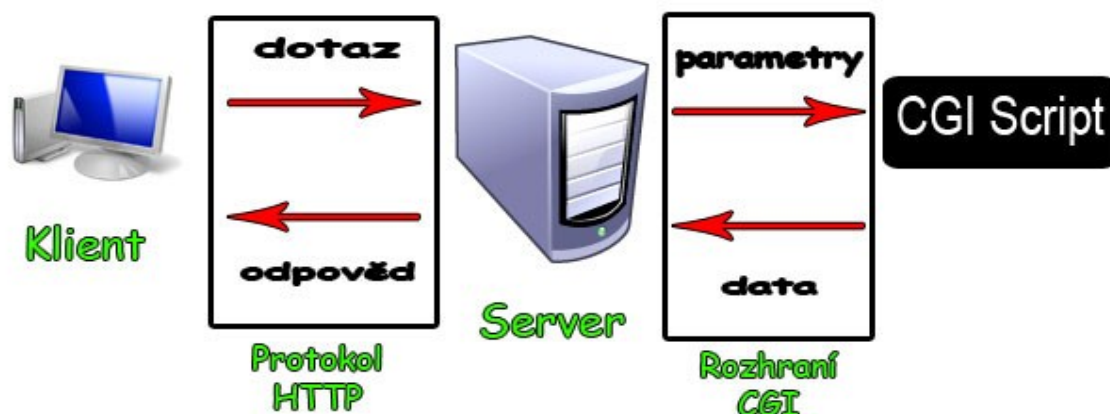
Určitě existuje větší množství standardů a prvků webových stránek, ale ty nás zatím nemusejí zajímat.

2.2 Vývoj webových aplikací

Na počátku internetu existovaly pouze statické textové stránky. Převládaly vědecké, odborné články, firemní internetové stránky poutaly převážně krátkým popisem předmětu jejich podnikání a kontaktními informacemi. Fotografie či obrázky se vyskytovaly v omezeném množství. Musely mít minimální datový objem, který omezil jejich velikost a rozlišení. Tento stav umožňovaly tehdejší technické možnosti Internetu. Rostoucí počet uživatelů a jejich potřeb, ale i rychlejší a kvalitnější připojení k internetu umožnili prostor pro vytváření objemnějších webových stránek s hodnotnějším obsahem. Postupem času vývojáři internetových stránek nahradily jednoduché formuláře HTML ke komunikaci s uživateli složitějšími a rozsáhlejšími skripty. Internet se tak začal stávat dynamičtější.

První používané skripty se nazývají CGI. Tyto skripty je možné psát v různých programovacích jazycích a používat k základním funkcím webových aplikací jako je ukládání uživatelských dat do databází, zobrazení dat z databází nebo generování HTML stránek na základě prováděných akcí uživatelem. Rozhraní CGI definuje způsob předávání dat mezi WWW-serverem a CGI-skriptem (Obr. 2.2.1). [31]

Předání dat mezi serverem a CGI skriptem



Obr. 2.2.1 – Způsob předávání dat mezi www serverem a CGI-skriptem

Jiný nástroj, kterým je možné tvořit dynamický obsah internetových stránek, se nazývá applet. Jedná se o softwarovou komponentu, která je napsaná v jazyce „JAVA“ a běží v kontextu jiného programu (většinou Java-kompatibilním prohlížeči). Nespouští se přímo (jako aplikace), nýbrž otevřením HTML dokumentu, kde je na něj umístěn odkaz pomocí speciální značky „<APPLET>“. Applety mohly přehrávat zvuk, volat veřejné metody appletů umístěných na téže webové stránce, požádat internetový prohlížeč o zobrazení libovolné www stránky. Applety mají z bezpečnostních důvodů některá omezení jako je např. zápis do systémových souborů, načtení knihovny, spuštění programů na domovském serveru nebo definování nativní metody. V době svého příchodu na internet se však netěšily velké oblibě, protože uživatele odrazovala nutnost mít nainstalovaný JVM, správnou verzi programovacího jazyka Java a další nepříjemnou vadou byla nepřiměřená doba načtení internetové stránky. Applet je totiž nutné před spuštěním celý stáhnout na klientský počítač. [32] Používanějším nástrojem pro tvorbu dynamických stránek je „JavaScript“. JavaScript není Java, jak by tomu mohl název napovídat, ale má pouze podobnou syntaxi jako zmíněný programovací jazyk. JavaScript je jazyk [33]:

- interpretovaný - nemusí se kompilovat
- objektový - využívá objektů prohlížeče a zabudovaných objektů
- závislý na prohlížeči - funguje ale ve většině prohlížečů
- case sensitivní - záleží na velikosti písem v zápisu
- syntaxí podobný jazykům C, Java a podobným

Jeho autorem je Brendan Eich z tehdejší společnosti Netscape. Je to klientský skript, což znamená, že se program odesílá ze serveru do počítače a teprve tam je zpracován. Nevýhodou JavaScriptu je funkčnost pouze v prohlížeči, velká složitost, uživatel jej může zakázat, existence rozdílných verzí, které ve spojení s různými verzemi prohlížečů mohou způsobovat značné problémy. I přes řadu těchto nevýhod je stále silným nástrojem pro vytváření dynamických webových stránek, hlavně díky širokému spektru možností využití. Při vývoji webových aplikací je potřeba myslet na to, že uživatel má možnost si jej ve svém prohlížeči zakázat. Vzniklou situace je nutné elegantním způsobem vyřešit. Jedna z nejznámějších a nejpoužívanějších javascriptových knihoven se nazývá „jQuery“. Obsahuje řadu funkcí, které usnadňují práci s javascriptem. Klade důraz na jednoduchost, čitelnost a rychlost. Je multiplatformní (funguje na více operačních systémech) a má pěkně zpracovanou dokumentaci i návody. JQuery umí:

- Události
- Manipulace s CSS
- Selektory
- Efekty (pomocí předdefinovaných funkcí)
- Animace - jednoduchá tvorba, velmi efektní i efektivní výsledek
- Ajax - načítání obsahu serveru bez nutnosti obnovení stránky
- Spoustu pluginů

JavaScript napomohl ke vzniku Document Object Model – DOM. Jedná se o objektově orientovanou reprezentaci XML nebo HTML (webové) stránky v podobě stromové struktury HTML tagů. Rozhraní umožňuje přístup či modifikaci obsahu, struktury, stylu dokumentu nebo jeho částí. DOM se stal standardem W3C a JavaScriptu se ujala asociace ECMA a zpřesnila jej pod označením ECMAScript. [30] Internetové stránky se skriptem, který byl vytvořen podle daných specifikací, by se měly zobrazovat a chovat stejně ve všech prohlížečích, které podporují tyto standardy. Jelikož i dnes není podpora prohlížečů úplně perfektní, mohou se stejné skripty chovat odlišně v různých browserech.

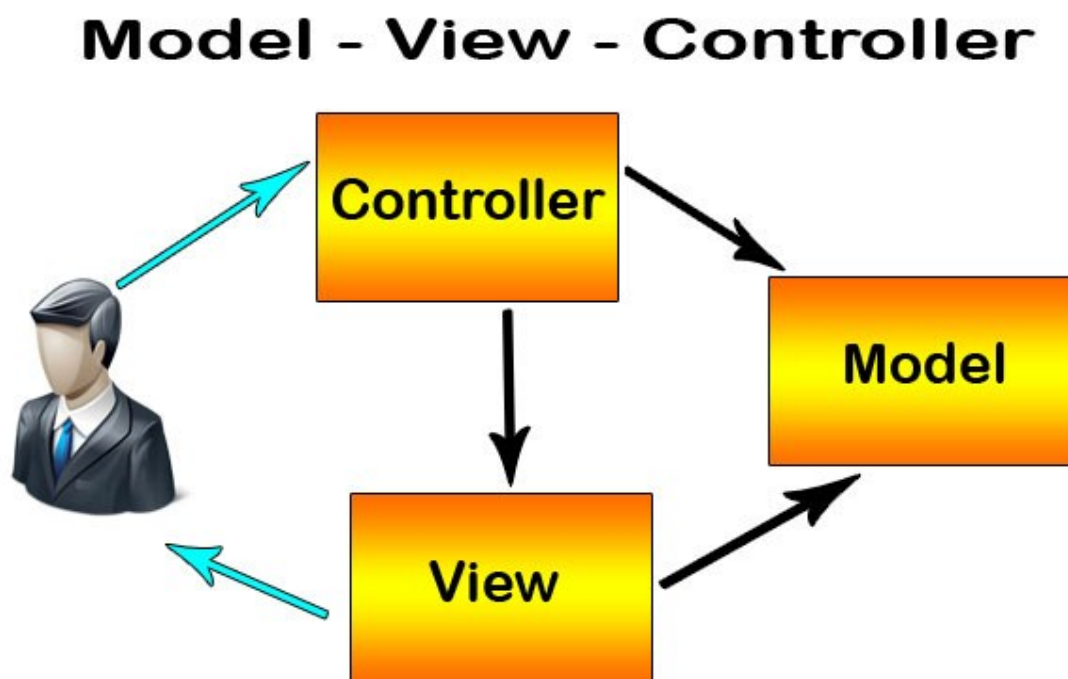
Pro potřebu tvorby dynamických webových stránek byla vyvinuta řada skriptovacích jazyků, které úzce spolupracují s webovým serverem. Mezi nejznámější a nejvyužívanější patří JavaServer Pages – JSP od společnosti Sun Microsystems (v dubnu 2009 byla prodána firmě Oracle), Active Server Pages – ASP od společnosti Microsoft (nástupcem je ASP.NET,

který lze chápat jako komplexnější technologii a od původního ASP se fundamentálně liší (v mnoha ohledech) a PHP od vývojářů PHP Group. Hlavní důvodem jejich vzniku je oddělit programovou logiku aplikace od vzhledu. Dnes mají tyto skriptovací jazyky řadu různých verzí, rozšíření, podporují objektově orientované programování a mohou sloužit k vytváření důmyslně propracovaných internetových aplikací. Další součástí dnešních webových stránek je technologie Flash, kterou vyvinula společnost Macromedia (v roce 2005 ji koupila společnost Adobe). Technologie Flash je určena především pro vizuální stránku webu. Umožní připravit internetovou prezentaci skutečně poutavou a na vysoké úrovni po grafické stránce. Hodí se zvláště pro prezentace, kde hlavní důraz je kladen na image a originalitu řešení. Lze ji využít pro jiné účely než je grafické oživení webových stránek, protože Flash v sobě skrývá i rozsáhlé možnosti programování, 2D a 3D animací. Pomocí skriptovacího jazyka ActionScript je možné vytvářet i hry. Flash dnes používá přes jeden milión vývojářů, kteří svými produkty mohou ihned oslovit více než 500 miliónů uživatelů internetu po celém světě. Nevýhodou technologie je, že uživatel musí mít nainstalovaný Flash Player, aby mohl spustit aplikaci. Další nevýhoda je, že vyhledávače neumějí indexovat (nalézt a popsat) obsah flash souborů. Webové prezentace by v žádném případě neměly být tvořené kompletně ve flashi. Nahrávání může trvat dlouho, uživatel bez flash playeru nic neuvidí a vyhledávač nenalezne relevantní obsah. Oproti vývoji appletů, jejichž zdrojový kód lze upravovat v jakémkoliv textovém editoru, je nutné k vytváření flash aplikací použít vývojové prostředí.

Další současně velice využívaný nástroj se nazývá Framework. Framework je programová struktura, která slouží jako pomoc při programování a vývoji jiných softwarových projektů. Může obsahovat podpůrné programy, knihovnu API, návrhové vzory nebo metodické postupy při vývoji. Hlavním cílem frameworku je usnadnit programování typických problémů dané oblasti. Příkladem může být vývoj webových stránek pro banku. Framework usnadní práci návrhářů a vývojářům tak, aby se soustředili především na problematiku bankovních operací nikoliv na zajištění bezchybné navigace mezi stránkami. Nejčastější oblastí, kterou se frameworky zabývají, jsou internetové obchody (e-shop). Vyskytují se námitky, že použitím frameworku bude kód pomalý či jinak neefektivní a že čas, který se ušetří použitím cizího kódu, se musí věnovat nastudování frameworku. Nicméně při jeho opakovaném nasazení nebo ve velkém projektu dojde k výrazné úspoře času. Při odinstalování frameworku již nebude možno některé aplikace spustit. Architektura se skládá z tzv. hot spots a frozen spots (doslova „horkých“ a „zmražených míst“). Frozen spots definují základní komponenty a strukturu frameworku a zůstávají stejná (zmražená). Hot

spots jsou komponenty definované pro funkcionalitu konkrétního vyvíjeného projektu ve frameworku a tedy jsou s každým projektem jiná.) Mezi nejznámější frameworky patří Zend Framework, Nette Framework, CakePHP, Symfony, Ruby on Rails a JUnit.

Zajímavým řešením dnešní doby při vytváření sofistikovaných webových aplikací je návrhový vzor (architektura) Model - View - Controller (dále jen MVC). Jedná se o architekturu, která rozděluje aplikaci na tři nezávislé komponenty (model, view, controller), aby je šlo upravovat samostatně a dopad změn byl na ostatní části co nejmenší. Model je v tomto případě nejjednodušší část, protože je identický s Modelem v desktopových technologiích. Obsahuje data a business logiku, s konkrétní prezentací nemá nic společného. O view se někdy říká, že u webové aplikace je to HTML, ale to je jako říci, že u desktopové aplikace se view rovná pixelům na obrazovce, respektive nízkou úrovním vykreslovacím instrukcím. Je ve skutečnosti serverový kód, který se o generování HTML stará, tj. PHP, C#, Java, Ruby a podobně. Navíc ani u webové aplikace není nutné, aby bylo výstupem HTML – klidně to mohou být formáty jako XML nebo JSON.

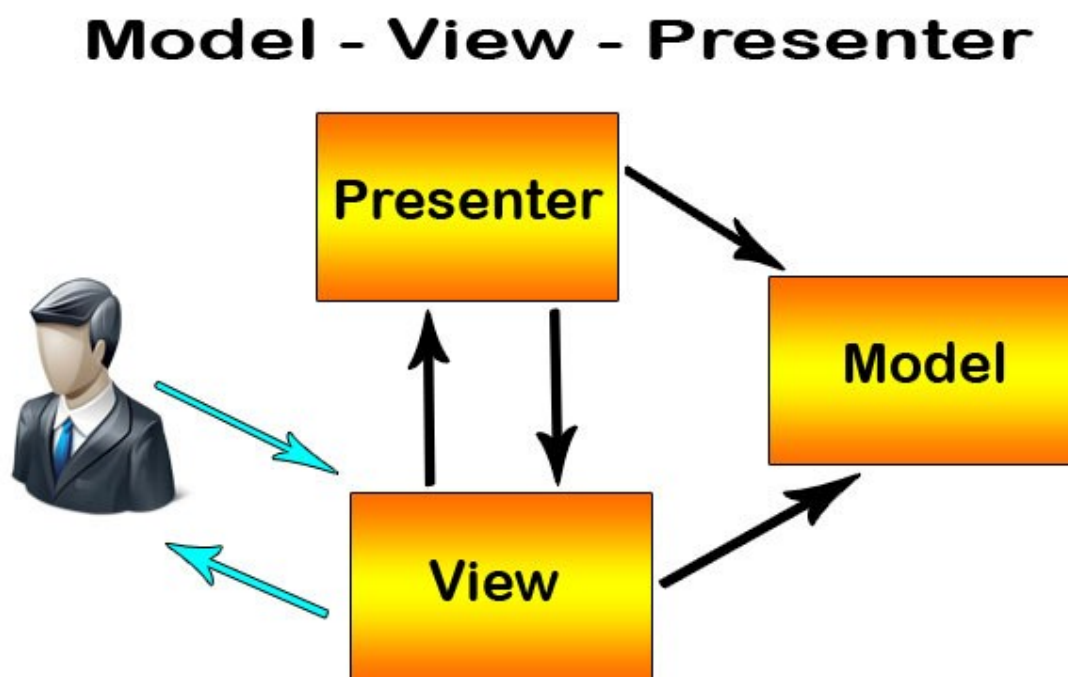


Obr. 2.2.2 - Architektura Model - Model - Controller

Tok událostí v aplikaci vypadá následovně (Obr. 2.2.2):

1. Uživatel vykoná nějakou akci na uživatelském rozhraní
2. Ta je zachycena controllerem
3. Controller rozhodne, jak na akci zareagovat, a typicky změní nějaké hodnoty v modelu nebo přímo ovlivní view
4. View zobrazí změny uživateli

Jak se programovací jazyky stále vyvíjely, input a output se postupně spojily do jednoho. Např. Button (tlačítko) dnes zvládá jak své vykreslení na obrazovku, tak ošetření události myši, klávesnice a dalších zařízení. Tím v podstatě odpadla potřeba „C - Controlleru“, který už nemusel ošetřovat vstup. Postupně se přišlo i na to, že oddělení aplikační logiky od definice uživatelského rozhraní je žádoucí, ačkoliv komponenty teoreticky mohou vstup ošetřovat samy. Vzniká tak velký bratr MVC - architektonický vzor Model – View – Presenter (dále MVP, obr. 2.2.3), kde Controller nahradil Presenter. Pro model MVP jsou přímo stvořené nové technologie RIA (Flex, Silverlight, JavaFX).



Obr. 2.2.3 – Model - View - Presenter

Rozdíly oproti modelu MVC:

- Na uživatelský vstup i výstup plně dohlíží View
- Oddělení view a presenteru není z nutnosti ošetření vstupu, ale čistě z architektonických důvodů. Prezentační vrstva se udržuje lépe než monolitické View
- Přímá vazba view na presenter.
- Presenter většinou pracuje přímo s View, tudíž i tato vazba je silnější než v případě vzoru MVC.

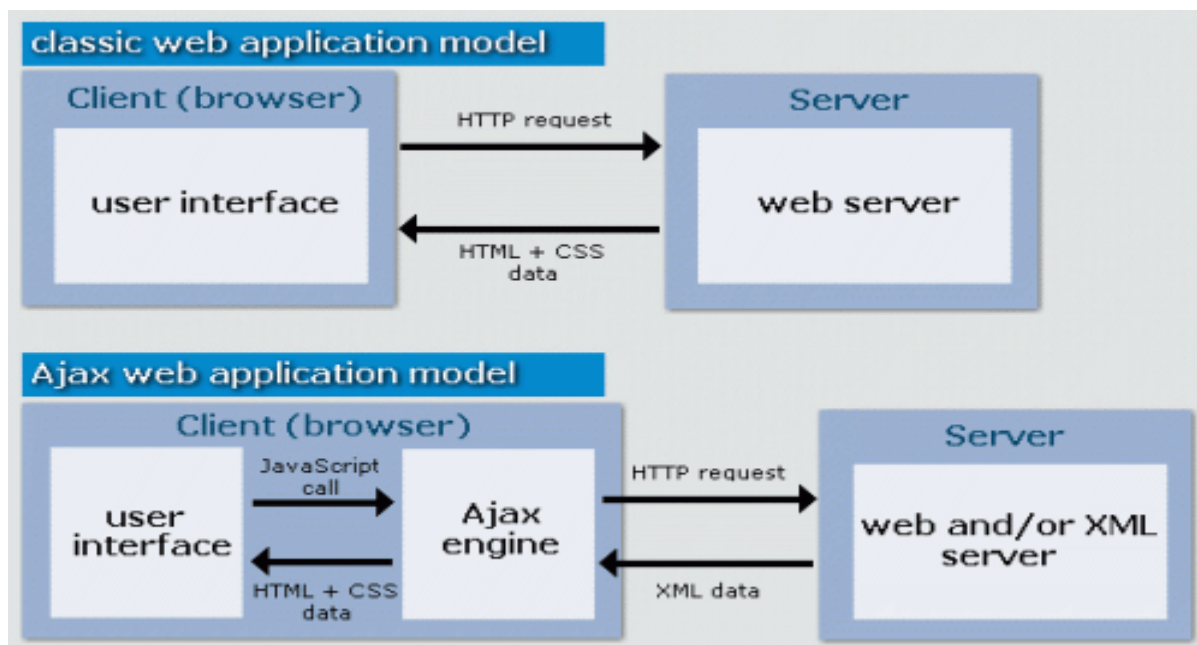
Změn je celkem dost, proto bychom si neměli namluvit, že vzory MVC a MVP se liší pouze posledním písmenem. [16]

Velký zájem na internetu vyvolalo dynamické HTML (dále jen DHTML). Nejedná se o standard, ale o spojení několika technologií používaných pro tvorbu dynamických a interaktivních webových stránek. Většinou se jde o technologie: HTML, JavaScriptu, kaskádových stylů (CSS – Cascading Style Sheets) a DOM. DHTML je protiklad ke statickému HTML, které se jednou načetlo do stránky a pak už se s ním nedalo nic dělat. Právě kombinace technologií umožňuje obsah stránky měnit i po načtení. Jsou to různé animace, změny textu, vyskakování oken, změny barev atd.. Kombinaci těchto technologií přinesly čtvrté verze prohlížečů (Netscape a Microsoft). I přes první rozpačité implementace pomohlo DHTML rozšířit a popularizovat technologie důležité pro budoucnost.

Poslední technologií, o které se zmíním, je AJAX. Zkratka AJAX znamená Asynchronous JavaScript and XML. Není sám o sobě implementací technologie či softwarovým produktem, ale jedná se o obecný koncept nebo lépe návrhový vzor pro bohaté internetové aplikace. Samozřejmě již dnes existují obecné implementace AJAX konceptu (viz Direct Web Remoting), které je možné integrovat do vlastního řešení. Je představitel cesty využívající maximální možné hodnoty dnešních technologií. Můžeme si jej představit jako pomyslný deštník, pod kterým se skrývají následující technologie:

- Document Object Model (DOM)
- XMLHttpRequest (Internet Explorer, Mozilla)
- HTML, CSS a JavaScript (JScript)

Možná si říkáte, co je na AJAXu tak převratného oproti klasickému webovému modelu. Je to koncept využití těchto technologií, a to především XMLHttpRequest k volání serveru. [9] Bližší srovnání klasického přístupu oproti AJAXu ilustruje následující obrázek (obr. 2.2.4):



Obr. 2.2.4 - Porovnání klasického konceptu s AJAXEM [12]

Základním stavebním kamenem je objekt XMLHttpRequest, který umožňuje asynchronní volání serveru. V klasickém webovém modelu každá změna stavu na klientu vyžaduje obnovení celého uživatelského rozhraní. Nejdříve je tedy žádost o změnu stavu, odeslání požadavku na server, vyřízení požadavku a vše končí zasláním kompletního uživatelského rozhraní s daty, přičemž jednotlivé kroky jsou vzájemně synchronizovány. Naopak AJAX, díky XMLHttpRequest, může vyvolat libovolný počet nezávislých požadavků, jejichž výsledky mohou ovlivnit pouze patřičné části uživatelského rozhraní, bez nutnosti jeho celkového znovu načítání. Tedy, žádost o změnu stavu, vygenerování požadavku přes XMLHttpRequest, vyřízení požadavku serverem a zpracování vrácené odpovědi XMLHttpRequestem a změna patřičné části uživatelského rozhraní. (Logika obsluhující XMLHttpRequest je na obrázku znázorněna jako AJAX engine.) [12]

Určitě existuje celá řada dalších technologií a nástrojů, které napomohly svým vývojem k dnešní podobě aplikací na internetu. Mým úkolem není čtenáře seznámit s těmito veškerými technickými řešeními, které se v současnosti i minulosti používají či používali.

Pouze poukázat na ty, které jsou z mého pohledu významné, dnes hojně využívané a napomohli k současné podobě internetových aplikací.

2.3 Druhy aplikací

Mnoho lidí si pod pojmem internetová aplikace představí program, stránky, které běží ve webovém prohlížeči jako je např. internetový e-shop (obchod), elektronickou poštu, internetovou aukci, bankovní informační server nebo hry. Každý možná něco slyšel o pojmech typu applet, HTML, JavaScript, CSS, ale neví přesně, o co se jedná. Jaký je vlastně rozdíl mezi webovou (internetovou) aplikací, RIA aplikací a desktopovou aplikací?

- Desktopová aplikace - software, který je napsán pro konkrétní operační systém (OS) a nainstalovaný výhradně nad tímto OS na jednom počítači. Pro zjednodušení uvažujeme aplikaci postavenou nad grafickým rozhraním toho OS. Právým opakem je webová aplikace
- Webová aplikace – je podle definice [13] klient/server software, který komunikuje s uživatelem nebo jiným systémem prostřednictvím protokolu HTTP. Za klienta používají uživatelé nejčastěji webové prohlížeče, jako je Internet Explorer, Mozilla nebo Opera. Automatizované systémy, například roboti fulltextových vyhledávačů, pak přistupují s pomocí HTTP agentů. Klient na základě interakce s uživatelem zasílá serveru jednotlivé požadavky a následně zobrazuje obdržené webové stránky zapsané zpravidla v jazyce (X)HTML. [13] Představují širokou možnost dynamicky generovat obsah internetových stránek na základě interakce s jejich návštěvníkem. Tato interakce může být jak velmi prostá – vyhledávání v textu, seřazení nabídky dle ceny, tak i velmi komplexní, např. sledování stavu zakázky s automatizovaným zasíláním upozornění, vlastní administrace obsahu webu, zasílání agregovaných dat pro další statistické zpracování internetovou aplikací, kniha hostů, online obchody, bankovní aplikace. Webová aplikace se tak pohybuje v rozmezí od uživatelsky orientovaného doplňku prezentace po roli aktivního prvku firemního informačního systému.
- RIA aplikace - jsou webové aplikace, které se snaží překlenout rozdíly mezi klasickou webovou aplikací a desktopovou aplikací. Snaží se v rámci webového prohlížeče napodobovat desktopové aplikace svým vzhledem i chováním a poskytnout vyšší uživatelský komfort.

2.4 Webové aplikace

Většina internetových aplikací má potřebu si data z Internetu vyžádat a následně zobrazit. Potřebuje způsob komunikace s uživatelem. Nejznámějším a běžně používaným způsobem komunikace je architektura klient - server.

2.4.1 Klient - server

Tato architektura patří mezi nejrozšířenější v oblasti TCP/IP sítí. Hlavní idea spočívala v tom, že zpracování dat má probíhat na stejném místě, kde jsou uloženy. Tímto místem se nazývá „server“, který poskytuje různé služby. Emailový server poskytuje služby týkající se přijímání a odesílání emailů. FTP server umožňuje sdílení souborů na síti nebo databázové servery sloužící ke správě a manipulaci dat uložených v databázi. Druhou aktivní částí architektury je zpravidla uživatelský počítač, zvaný „klient“, který naváže komunikaci se serverem, pošle mu potřebná data a obdrží výsledek. Existují dva základní druhy klientů:

- **Thlustý klient (thick client)** – obsahuje v sobě jak presentační tak i aplikační vrstvu a připojuje se přímo k databázovému nebo jinému serveru. Instalace, konfigurace a podpora těchto klientů je finančně a časově náročná. Zároveň klade větší nároky na hardware a software na uživatelském počítači.
- **Tenký klient (thin client)** - je obvykle webový prohlížeč, který s presentační vrstvou komunikuje přes bezstavový HTTP protokol a stará se tak pouze o zobrazování dat. Neprobíhá zde žádná rozhodovací logika, pokud nebereme v úvahu např. validaci dat, zadávaných do webového formuláře. Z výše uvedené skutečnosti je zřejmé, že uvedené nároky u předchozího klienta jsou mnohem nižší. Vývoj aplikace pro tenkého klienta je náročnější především z důvodu rozčlenění aplikace do jednotlivých vrstev. Tenký klient se dělí na softwarového a hardwarového klienta. Softwarový tenký klient je svým konceptem podobný tenkému klientovi hardwarovému, s tím rozdílem, že prostředí pro běh aplikace je dáno softwarovou aplikací provozovanou na klasickém počítači - tlustém klientovi.

Existují i další dva klienti. Jedná se o chytrého klienta (smart client) a hybridního klienta. Chytrý klient je takový typ aplikace, která je podobně jako u tenkého softwarového klienta poskytována serverem. Rozdíl je v tom, že aplikace samotná je klientem a je provozována na hardwarovém vybavení, které může jako lokální zdroje využívat. Protože aplikace je kompletně spouštěna na klientském počítači, může být do jisté míry nezávislá na síťové konektivitě. U hybridního klienta se jedná o hardwarové vybavení obdobné tlustému klientovi, ale nedisponuje vlastním pevným diskem. Veškerý software pro běh systému a

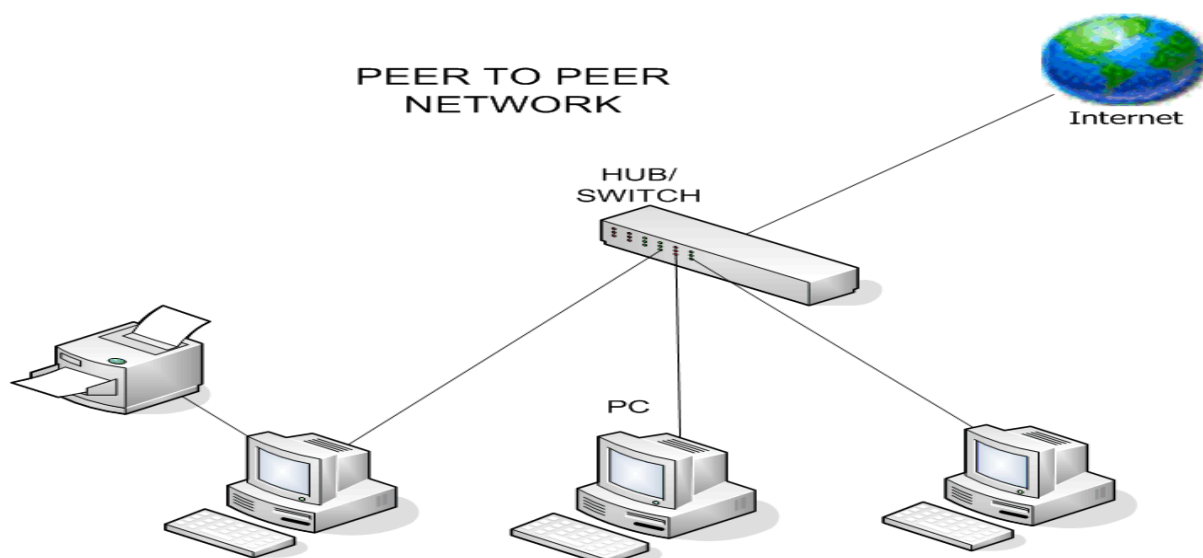
aplikace je získáván ze serveru, což umožňuje jeho centralizovanou správu. Při samotném běhu aplikace je již možné využívat lokální zdroje, stejně tak jako síťovou konektivitu a výkon serveru. [18] Architektura klient-server zajišťuje především tři základní činnosti:

- Prezentační činnost - komunikace s uživatelem přes grafické rozhraní (např. zobrazení výsledku na základě požadavku uživatele)
- Poskytování aplikační logiky – samotná funkcionality poskytování služeb
- Správa dat – zahrnuje mazání, úpravu, kontrola persistence dat atd.

Dalším používaným způsobem komunikace je peer-to-peer (rovný s rovným), zkráceně P2P. Je pravým opakem výše zmíněné architektury.

2.4.2 Peer-to-peer

Jedná se o síť, pomocí které lidé sdílejí a vyměňují nejrůznější soubory, jako jsou videa z mobilu, studijní materiály, fotografie z dovolené. Často se ale prostřednictvím těchto sítí nelegálně sdílí programy, hry, hudba a filmy. I proto je P2P komunikace trnem v oku ochráncům autorských práv. P2P komunikace nevyžaduje žádný speciální server, na kterém by tato služba běžela. Počítače v síti jsou naopak schopné napojovat se jeden na druhý, vyměňovat si data. Uživatelé tedy dostávají prostřednictvím P2P softwaru možnost své soubory sdílet a současně je z jiných počítačů stahovat. Mezi nejznámější programy patří např. uTorrent, BitTorrent, Strong DC atd. Jednou ze základních výhod P2P sítí je fakt, že s rostoucím množstvím uživatelů celková dostupná přenosová kapacita *roste*, zatímco u modelu klient-server se musí uživatelé dělit o konstantní kapacitu serveru, takže při nárůstu uživatelů klesá průměrná přenosová rychlost.



Obr. 2.4.2 – Komunikace Peer - To - Peer

3. Trendy vývoje robustních internetových aplikací

3.1 Zkratka RIA

V další části diplomové práce se budu zmiňovat o technologiích RIA – Rich Internet Application. Doslovný překlad z angličtiny do mateřského jazyka znamená: „Bohatá internetová aplikace“, tedy v jednotném čísle jedna aplikace. Význam pojmu RIA se postupem času vyvinul vpřed a už neznamena jen aplikaci. Zkratka RIA označuje vývoj (směr), kterým se ubírají současné programy na Internetu. Proto je nutné rozlišovat správné použití pojetí RIA. Definuji-li konkrétní program nebo technologii, budu používat spojení „RIA aplikace“, resp. „RIA technologie“. V následujících odrážkách je uvedeno upřesnění, které budou v textu dále používány.

- RIA technologie - souhrn technologických postupů a prostředků pro vytvoření RIA aplikace. Jedná se především o název technologie, základní programovací jazyk a koncept komunikace klient - server.
- RIA aplikace - vytvořená nad konkrétní RIA technologií.

V ostatních případech budu používat kontinuitu „pojem RIA“. Hlavní snahou je se vyhnout samotné zkratce RIA bez přívlastku, aby nedošlo k nepřesnému pojmenování.

3.2 Definice pojmu RIA

Než se dostanu k podrobnému popisu pojmu RIA, je nutné vymezit tento pojem do takového rozsahu, abychom byli schopni mít alespoň určitou představu, co bude náplní mé diplomové práce. RIA je pojem, který svou obecností dokáže vyděsit nejednoho web developera, proto začnu konkrétními příklady: GMail, Zoho Writer, Outlook Web Access, Buzzword, Photoshop Express, Sliderocket jsou RIA a řada dalších aplikací. Tak by šlo pokračovat pro tisíce AJAXových aplikací, které jsou dnes na webu k nalezení. Takže i když jste třeba pojem RIA nikdy neslyšeli, některou z těchto aplikací jste už určitě používali. Rozhodně však nesmíme chápat, že AJAX = RIA. Nejjednodušší varianta definice by mohla být následující: „RIA (Rich Internet Application) jsou webové aplikace, které se snaží překlenout rozdíly mezi klasickou webovou aplikací a desktopovou aplikací. RIA aplikace se snaží v rámci webového prohlížeče napodobovat desktopové aplikace svým vzhledem i chováním a poskytnout vyšší uživatelský komfort.“ [10]

Jak je z definice zřejmé, tak RIA aplikace mají stejné znaky (funkce) jako klasické softwarové programy. Těmito vlastnostmi jsou nejčastěji chápány ve smyslu uživatelského

rozhraní GUI (Graphical User Interface). Na první pohled by se dalo o desktopových aplikacích mluvit jako o Rich Applications (Bohaté aplikace) bez přívlastku Internet. Tento český výraz není moc výstižný, protože v praxi se o desktopových programech nemluvilo jako o bohatých aplikacích. Důvodem tohoto označení může být vztah mezi klasickými programy a internetovými aplikacemi, které v porovnání s nimi působí nižší kvalitou ve významu nabízených schopností. Výše zmíněná definice Rich Internet Applications je příliš obecná a neurčitá, ale nikoliv nepravdivá. Svědčí to o tom, že hledat definici něčeho, co nemá platný a rozšířený standard, je velmi obtížné. Pokud si projdeme články nebo diskuze, kde je zmínka o pojmu RIA, většinou dojde k velké diskuzi a k ostré výměně názorů o správné formulaci tohoto významu. Hlavní důvod je ten, že koncept RIA pokrývá širokou oblast a je velmi těžce definovatelné.

3.3 Počátky Rich Internet Applications

Pojem „Bohaté internetové aplikace“ byl představen v březnu roku 2002 společností Macromedia (nyní společnost Adobe). Společnost tehdy vydala dokument „white paper“ (autorem je Jeremy Allaire), v němž byl poprvé použit pojem RIA jako kombinace „Rich Client“ a „Internet Application“. Úvodní část dokumentu obsahuje ohlédnutí za vývojem webových aplikací od roku 1990, dále následuje vývoj a využití bohatých internetových aplikací a vše předvádí na svém produktu Macromedia Flash MX. Dodnes je technologie Flash jedna z nejlepších v oblasti RIA technologií a pomáhala ji spoluvytvářet. Co vlastně RIA technologiím předcházelo? Ještě předtím než se dostaneme k samostatným bohatým internetovým aplikacím, zmíním se o myšlenkách (technologiích), které předcházeli rozšíření pojmu RIA a snažili se o přiblížení světů – desktopových a webových aplikací.

3.3.1 Remote scripting – vzdálené skriptování

V roce 1990 vymyslela firma Microsoft technologii „Remote Scripting“, která skýtá nástroje pro odstranění opakovaného načítání celé HTML stránky při klientském dotazu na server. [11] Technologie umožňuje volat vzdálené serverové funkce z klientského počítače, jako kdyby byla volána a zpracována funkce přímo na straně klienta. Technologie Remote Scripting využívá tři klíčové komponenty:

- Java Applet – aplikovaný na straně klienta
- Microsoft JScript – aplikovaný na straně klienta
- JScript – použitý na straně serveru

Webová stránka, používající vzdálené skriptování, obsahuje skript, jehož funkce zjistí, jaký internetový browser je využíván a následně načte Java applet, který je hlavním elementem v komunikaci klient – server. Applet plně zprostředkuje veškeré požadavky klientů na server. Server obdrží dotaz od klienta, který pošle právě Java applet, server zpracuje parametry a zpět Java appletu na straně klienta předá výstupní hodnoty ve formátu XML, které se následně zpracují a doplní do HTML stránky. Tato komunikace může probíhat synchronně (klient počká na výsledek operace na serveru) nebo asynchronně (klient na výsledek nečeká, ale může se o ukončení operace na serveru dovědět prostřednictvím události). Pro realizaci spojení mezi stranou klienta a serveru je nutné mít nainstalovány následující soubory:

- *Rs.Htm* – Obsahuje metody nutné k inicializaci komunikace na straně klienta
- *Rs.Asp* – Obsahuje metody potřebné na straně serveru
- *Rscopy.Class* – Třída formuluje a plní roli Appletu, který uskutečňuje spojení mezi klientem a serverem

Následující ukázkový kód obsahuje dva klíčové řádky, které realizují Remote Scripting na straně klienta.

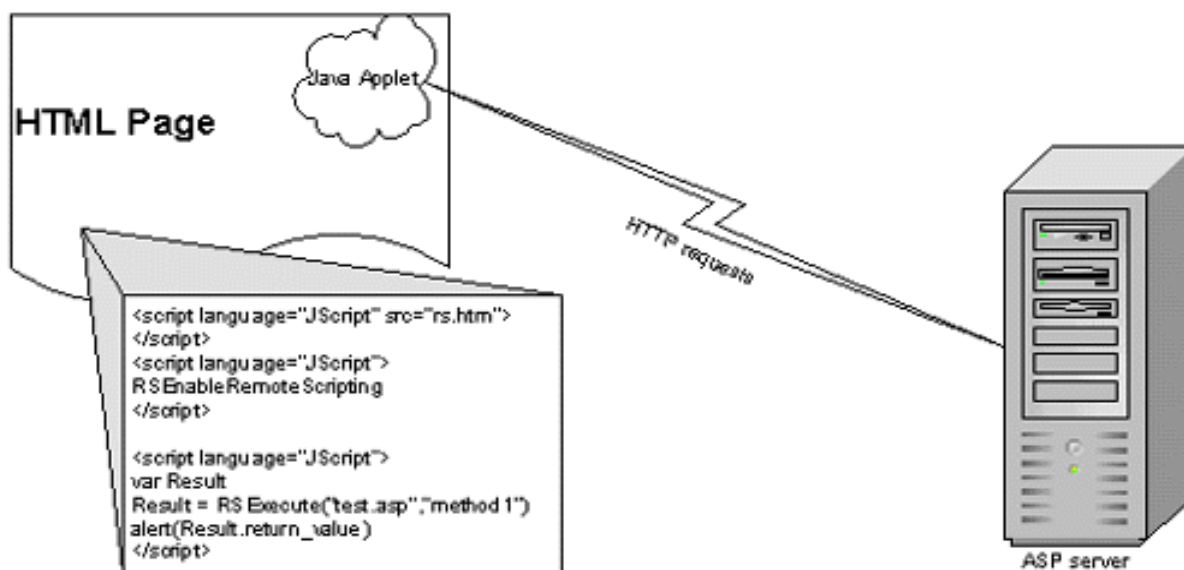
- `<SCRIPT LANGUAGE="JavaScript" src="../_ScriptLibrary/Rs.Htm"></SCRIPT>`

Tento řádek vloží do stránky klienta knihovnu, která umožní vzdálené skriptování.

- `RSEnableRemoteScripting(../_ScriptLibrary)`

Volání této metody zajistí inicializaci Remote Scripting knihovny.

Celý komunikační proces je znázorněn na obrázku 3.3.1, který je spolu s popisem technologie převzatý ze zdroje [11].



Obr. 3.3.1 - Komunikační proces Remote scripting [11]

3.3.2 X-Internet

V roce 2000 zveřejnil George Colony z Forrester Research svou vizi o budoucnosti Internetu. Tvrdil, že doba Internetu založeného na tehdejších webových technologiích je u konce a zanedlouho budou vystřídány novými, které přinesou zcela nový postoj k využívání Internetu. Novou epochu Internetu nazývá X-Internet [4]. Písmeno X má dva klíčové významy:

- eXecutable Internet – spustitelný Internet
- eXtended Internet – rozšířený Internet

První význam, tedy eXecutable Internet je zde použit ve smyslu, že by mělo být umožněno vykonávat alespoň částečně aplikační logiku na straně klienta (použití konceptu „Smart Client – chytrý klient“ namísto konceptu „Thin Client – tenký klient“). Takže logika je symetricky rozdělena mezi klienta a server, čímž je dosaženo snížení vytíženosti sítě a zároveň se tím zkrátí doba odezvy. Používání aplikace je tím pádem pohodlnější a jednodušší. Druhý význam, tedy eXtended Internet je chápán jako propojení fyzického a online světa, konkrétněji přístupu k Internetu z mnoha nejrozličnějších koncových zařízení a jejich vzájemnou komunikaci a spolupráci. Koncovým zařízením může být mobilní telefon, zařízení opatřené RDIF (technologie radiofrekvenční identifikace), automobil, domácí spotřebiče atd.. Ve firemní sféře by se dal tento koncept využít při online monitorování výrobků v rámci výroby či logistiky.

V dnešní době už téměř nenarazíme na pojem X-Internet. Je nutné připomenout, že ve své době byly tyto myšlenky velmi nadčasové a zejména v souvislosti s eXecutable Internet jsou v nynější době naplňovány v aplikacích pojmu RIA. [4]

3.4 Oblasti působnosti RIA aplikací

V této části se pokusím vymezit oblast působnosti bohatých internetových aplikací. Je nutné rozlišit RIA aplikaci od klasické desktopové, abychom měli jasno, co si pod každou můžeme představit. Pod pojmem „Desktopová aplikace“ si představme software, který je nainstalovaný na konkrétním operačním systému a nemá pevnou vazbu na internetové zdroje. Jeho funkcionality je nezávislá na internetovém připojení. Příklady takových aplikací jsou např. kancelářská sada - Microsoft Office, grafický editor – Adobe Photoshop, přehrávač multimédií – Microsoft Media Player a spousta dalších programů. Spousta lidí namítne, že i k těmto aplikacím je potřeba Internetu. V dnešní době je už samozřejmostí, že každý kvalitnější software před finálním spuštěním kontroluje aktuálnost nainstalované verze programu nebo v případě uživatelského dotazu vyhledává nápovědu na Internetu atd. Rozhodně je však nemůžeme označit za RIA aplikace, protože jejich hlavní funkcionality nestojí (je nezávislá) na internetovém připojení. Desktopové aplikace mají blíže k aplikacím typu Tlustý klient. Jak jsem si již dříve uvedl, tak ani internetové aplikace nejsou RIA aplikacemi. Internetové aplikace jsou bližší aplikacím typu Tenký klient. Dále bychom měli rozlišit RIA aplikace podle zpracování aplikační logiky:

- Klientsky orientované RIA aplikace – zpracování aplikační logiky převládá na straně klienta v případě Tlustých klientů
- Serverově orientované RIA aplikace – zpracování aplikační logiky převládá na straně serveru v případě Tenkých klientů

Na základě těchto omezení (rozlišení), které jsem právě uvedl, mohu dospět k určitým závěrům, které nastíním v následujícím textu. Opět se vrátím k tomu, že pojem RIA má velice široký význam a není jej vůbec lehké definovat. V některých případech může dojít k polemice, zda se opravdu jedná o RIA aplikaci. Původně se vztahoval pouze k vlastnostem technologie Flash, ale v současnosti se spíše posunul k významu chápání aplikací. Rich Internet Applications kombinují vlastnosti jak webových aplikací, tak desktopových aplikací. RIA aplikace nejsou jen webové aplikace, ale v některých případech i desktopové aplikace. Takovým příkladem může být aplikace eBay Desktop. Určitě mnoho lidí už někdy

nakupovalo přes internet. Právě „eBay“ je jeden z nejrozšířenějších on-line obchodů na našem kontinentě a možná i na světě. Aplikace eBay Desktop je nástrojem založeným na Adobe AIR, který přímo z naší plochy umožní svůj účet v tomto internetovém obchodě spravovat. Nainstaluje se na osobní počítač, spustí se v požadovaném běhovém prostředí (zmíněné Adobe AIR), ale bez přístupu k internetu by ztratila smysl. Následující příklad naznačuje, že v některých případech je opravdu složité určit, zda se jedná o klasickou desktopovou nebo už o webovou aplikaci. Snad se v blízké době dočkáme striktního rozlišení mezi těmito aplikacemi. RIA aplikace by se měla vyznačovat lehkou dostupností, zaváděním, údržbou a měla by být multiplatformní (nezávislá na operačním systému). Dnes je zcela běžné, že lidé v mnoha státech disponují mobilním zařízením. Právě pro takové zařízení existují implementace stávajících RIA technologií (např. technologie JavaFX). Tato oblast je zatím především z výkonnostních důvodů v počátcích, ale pojem RIA budeme slyšet i tam. Díky souměrnému rozdělení aplikační logiky mezi serverem a klientem, máme možnost pracovat s aplikací i v offline provozu (přítomnost skriptování na straně klienta). V offline režimu nejsou dostupné všechny funkce, ale pouze zlomek z nich. Tuto vlastnost oceníme hlavně v případě nečekaného výpadku internetového připojení nebo v rámci mobilních zařízení.

3.5 Rich Internet Application

Ve 21. století se internetové stránky, běžící na standardech Internetu, ubírají novým směrem, který se nazývá Rich Internet Applications, zkráceně RIA. Hlavní snaha spočívá v přizpůsobení vzhledu a chování desktopovým aplikacím. Anglické slovíčko „Rich“ přeložené do českého významu znamená: Rich = Bohatý. Ve smyslu webových aplikací se jedná o souhrn vlastností např. efektivní, interaktivní, flexibilní, rychlý, dostupný atd.. [13] Jednotlivé RIA aplikace můžeme rozdělit podle účelu vytvoření a podle prostředí pod kterým aplikace poběží do čtyř základních oblastí. Obrázek 3.5 znázorňuje samostatné oblasti, jedná se o web/html, pluginy prohlížečů, desktop a mobilní zařízení.



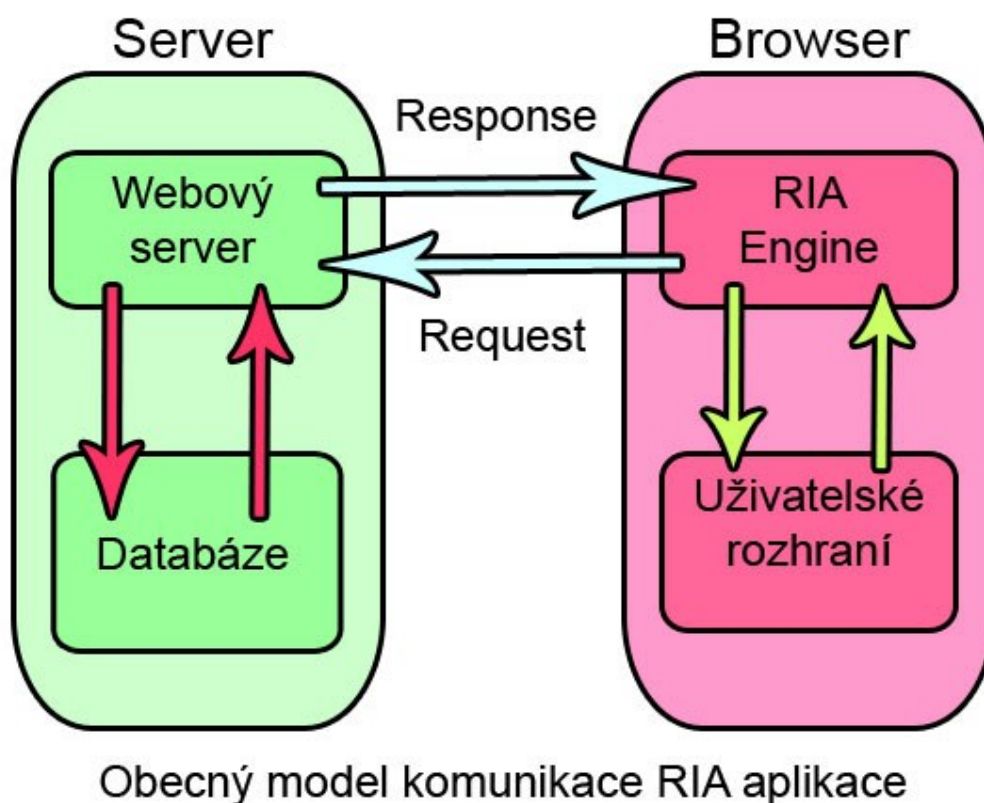
Obr. 3.5.1 – Oblasti RIA aplikací [13]

Z pohledu zisku je nejatraktivnější oblast webových aplikací. Pravděpodobně z tohoto důvodu se nejvíce firem (společností) zaměřilo na vývoj RIA aplikací právě pro konkrétní oblast a díky tomu existuje množství dostupných technologií v rámci daného pole působnosti. Tabulka 3.5.1 ukazuje nejznámější technologie.

Aplikace pro mobilní zařízení	Desktop aplikace	Webové aplikace
Oracle JavaFX Mobile	Windows Presentation Foundation (WPF)	Adobe Flex
Adobe Flash Lite	Adobe AIR	Microsoft Silverlight
		Google Web Toolkit
		Oracle JavaFX
		OpenLaszlo

Tab. 3.5.1 – Nejznámější technologie v rámci dané oblasti

V rámci své diplomové práce se budu zabývat oblastí webových aplikací, ostatní oblasti nebudou dále podrobněji popisovány. Jak sem již několikrát zmínil, tak RIA aplikace představuje kombinaci desktopových a webových aplikací. Obrázek 3.5.2 znázorňuje obecný model komunikace RIA aplikace, která je postavená na klasickém modelu webových aplikací a je rozšířena o klientskou vrstvu. Umožňuje zpracování požadavku na straně klienta a je umístěna mezi logickou a prezenční vrstvou.



Obr. 3.5.2 – Obečný model komunikace RIA aplikace

Řada technologií nabízí množství grafických prvků, které byly ještě donedávna dostupné pouze v desktopových aplikacích, aby dosáhly přívětivějšího, jednoduššího a uživatelsky příjemnějšího prostředí.

3.6 Vlastnosti RIA aplikací

Zásadní rozdíl oproti webovým aplikacím je v komunikaci request-response (požadavek-odpověď). Ačkoliv obě aplikace používají ke komunikaci se serverem výše zmíněný princip, RIA aplikace používá jinak. Klientská aplikace nekomunikuje přímo se serverem. Užívá prostředí, ve kterém běží jako zprostředkovatel (klientský engine), který přeposílá požadavky (dotazy) klienta na server a zpracovává odpovědi. Důležité je zmínit, že vše probíhá asynchronně, nikoliv synchronně. Na začátku si klientský počítač natáhne webovou aplikaci a následně je zobrazena na monitoru. V tomto okamžiku nebrání nic uživateli, aby mohl začít pracovat s aplikací. Místo přímého volání serveru (oproti synchronní komunikaci), tak RIA aplikace požádá o data klientskou vrstvu, která vytvoří požadavek a zašle ho serveru. Celá aplikace se nemusí načítat neustále znovu po každém novém

požadavku. Klientská vrstva si pamatuje všechny dotazy a podle jednotlivých odpovědí serveru aktualizuje potřebné komponenty. Uživatel celou komunikaci nevidí a po dobu jednoho volání serveru může pracovat s jinou částí aplikací. Tento způsob komunikace se využívá při zpracování většího množství dat z databáze.

3.7 Výhody a nevýhody RIA aplikací

Rich Internet Applications nabízí řadu výhod a důvodů pro jejich používání. Mezi ně patří bezesporu výše zmíněná asynchronní komunikace, při které se snaží optimalizovat přenos dat mezi klientem a serverem a tím zrychlit samotnou komunikaci. Další výhodou je, že nabízejí komfort funkcí a líbivější prostředí oproti klasickým webovým řešením (drag&drop, klávesové zkratky, kontextová nápověda). Je mnohem snazší začlenit RIA aplikaci do internetové stránky. Na lokálním počítači musíme mít nainstalovaný prohlížeč a zásuvný modul, který se stará o běh aplikace. Není potřeba žádné údržby, protože aplikace se zcela sama aktualizuje při opětovném navštívení stránky. I když se na první pohled zdají RIA aplikace jako bezproblémové řešení, pár nevýhod by se našlo. Jednou z nich je povolení JavaScriptu v prohlížeči klienta a případná instalace prostředí pro bezproblémový běh aplikace. Jinou nevýhodou je rychlost stahování dat a tím související závislost na kvalitním internetovém připojení. Mezi další nevýhodou můžeme zařadit neviditelnost obsahu pro internetové vyhledávače. Samotné aplikace jsou distribuovány přes internet, je zde tudíž závislost na internetovém připojení. I když se některé RIA technologie snaží tuto nevýhodu odbourat, stále je vyžadováno alespoň příležitostné připojení. Mezi další nevýhody můžeme rovněž zařadit rychlost stahování a zpracování skriptu na straně klienta. Určitou nevýhodou je také uzavřenost implementační vrstvy vůči okolí, ta může omezovat přístup k lokálním zdrojům. S uzavřeností aplikací je spojena také další nevýhoda, kterou je neviditelnost obsahu aplikace pro internetové vyhledávače. Nasazení RIA aplikací v podniku přináší hned několik výhod. Poutavé, informativní, samoobslužné aplikace Rich Internet Application povzbudí vaše zákazníky, aby zůstali déle na vašem webu, objevili více možností, provedli více online transakcí, utratili víc peněz, doporučili web dalším lidem, častěji se vraceli na web atd.. Dobře navržené aplikace mohou: [35]

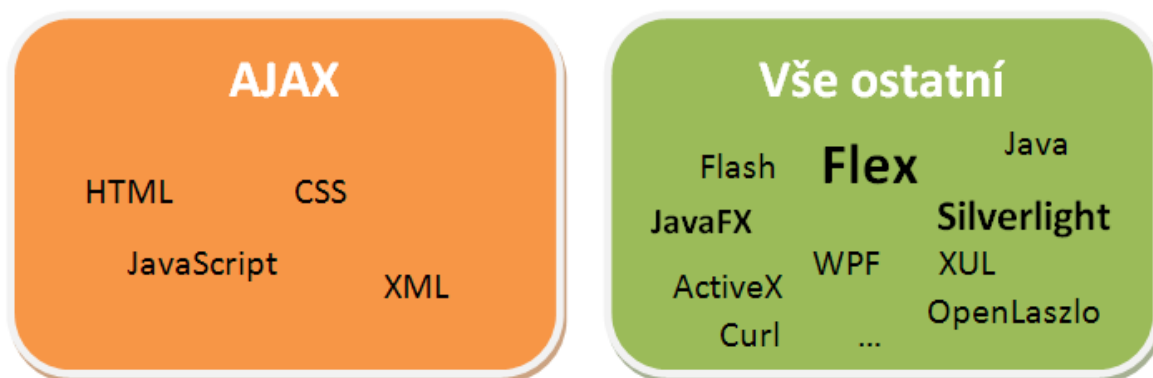
- zvýšit prodej online až o 30 %
- zvýšit množství převodů, mezikanálových prodejů a následného prodeje až o 40 %
- Snížit volání do kontaktních center/telefonním agentům ročně až o 20 %

Interní použití aplikace RIA mohou podnikům pomoci v následujících oblastech: [35]

- Sjednocení klíčových dat z různých zdrojů
- Zvýšení produktivity zaměstnanců
- Snížení doby zadávání dat až o 40 %
- Zvýšení přesnosti
- Úspora času
- Snížení provozních výdajů

3.8. Současné RIA technologie

V současnosti existuje celá řada technologií, které jsou určeny pro tvorbu bohatých internetových aplikací. Současně s vývojem těchto aplikací se spustil veliký boj společností o dosažení největšího podílu na trhu v rámci oblasti RIA aplikací (technologií). Pokud se v současnosti rozhodnete začít vývoj aplikací s přívlastkem RIA, máte na výběr dvě cesty. Půjdete cestou proprietární technologie, nebo cestou maximálního využití stávajících technologií a prostředků. Obě cesty jsou možné a již dnes lze najít úspěšná řešení, která se vydala jedním či druhým směrem. První cesta k realizaci těchto aplikací je obecný koncept - AJAX. Druhou cestou jsou aplikace, které využívají vlastní prostředky a technologie jako Flash, Java, vlastní programovací či skriptovací jazyk.



Obr. 3.8 – Obecné rozdělení RIA aplikací [13]

3.8.1 AJAX versus zbytek světa

AJAX využívá starých a dobře zavedených technologií, které jsou velice rozšířené a oblíbené, že je dnes podporují všechny prohlížeče (HTML, CSS, JavaScript, XML). Kromě všeobecné rozšířenosti, která je vynikající, existují spíše už jen určité problémy:

- Technologické schopnosti HTML/CSS jsou omezené výhradně na zobrazení textu
- Výkonnost JavaScriptu je slabá
- Podpora AJAX vývoje v nástrojích je slabá v porovnání s ostatními technologiemi
- Odlišnosti JavaScriptu oproti klasickým programovacím jazykům typu JAVA nebo C# vytváří překážky pro zkušené vývojáře s programováním serverů a desktopových aplikací
- Není snadné každou aplikaci odladit tak, aby běžela a vypadala stejně na všech internetových prohlížečích.

Díky těmto problémům vzniklo mnoho technologií, které se od sebe vzájemně liší, ale spojuje je jedna věc. Pro jejich běh je potřeba zásuvného modulu (plugin) do prohlížečů nebo jiné běhové prostředí. Seznam těchto technologií se nachází v pravé části obrázku 3.8.

3.8.2 Zbytek světa

Jak již bylo zmíněno, radí se sem technologie, které pro svůj běh na straně klienta potřebují instalaci běhového prostředí. Příkladem takového prostředí může být Flash Player od firmy Adobe. Běhové prostředí jsou v podobě zásuvných modulů pro internetové browsery a samotná instalace je velice jednoduchá, rychlá a intuitivní. Tato skupina je představena řadou technologií, z nich nejznámější jsou např.: Adobe Flex, Microsoft Silverlight a JavaFX. Jelikož oblast RIA není úplně přesně vymezená, dá se do ní zařadit i mnoho dalších technologií jako Google Web Toolkit, XUL, Mozilla Prism, Curl, OpenLaszlo atd. Některé jsou v principu podobné Flexu nebo Silverlightu, čímž jejich šance na úspěch je značně omezena, zatímco jiné se snaží řešit poněkud jiné scénáře. Google Web Toolkit přidává do typicky webových aplikací podporu práce v režimu offline.

3.9 Shrnutí

Nebýt dvou oblastí (desktop a mobilní zařízení), daly by se RIA aplikace definovat jako výše uvedená definice: „RIA (Rich Internet Application) jsou webové aplikace, které se snaží překlenout rozdíly mezi klasickou webovou aplikací a desktopovou aplikací. RIA aplikace se snaží v rámci webového prohlížeče napodobovat desktopové aplikace svým vzhledem i chováním a poskytnout vyšší uživatelský komfort“. Díky expanzi na desktop a do mobilních zařízení se však sám pojem „webová aplikace“ dostává do určitého sporu, proto je dnes vhodné RIA považovat za směr než za něco, co by šlo exaktně definovat.

4. Srovnání vybraných RIA technologií

Následující kapitola obsahuje detailní představení pěti nejznámějších technologií. Pole RIA technologií je pro vývoj bohatých (robustních) internetových aplikací skutečně široké, vzhledem k velkému množství různých technických řešení jsem vybral pět zástupců RIA technologií, které jsou dobře známé, široce používané, mají velkou pozornost mezi vývojáři, slibují zajímavé možnosti nebo v některých aspektech vynikají nad ostatními. Tvoří kombinaci různých technologií, programovacích jazyků a metod vytváření aplikací. Pomocí jejich analýzy si bude možné vytvořit ucelenější představu o soudobých možnostech rozvoje RIA aplikace. Pořadí hodnocení, které jsem zvolil, nemá žádnou souvislost s jejich popularitou nebo mojí oblíbeností k těmto technologiím. Pořadí je voleno náhodným výběrem. Nejprve se zaměřím na výběr technologií, které budu srovnávat. Následně pak sestavím metodiku analýzy jednotlivých technologií – tedy kritéria, na které se zaměřím ve své analýze. Před samotným srovnáním ještě detailněji formuluji metodiku bodového hodnocení, kterou poté aplikuji na jednotlivá technická řešení. Výsledkem bude ohodnocení, které nám poskytne určité srovnání technik pro tvorbu robustních internetových aplikací. Ještě než se pustím do samotné analýzy, musím upozornit čtenáře, že srovnání nebude mít absolutní váhu, protože použití dané technologie bude vždy záviset na konkrétním případě.

4.0.1 Zvolený výběr RIA technologií

Velké softwarové společnosti odhalily, že oblast RIA aplikací je v dnešní době marketingově velmi zajímavá a začaly se ji naplno věnovat. Z toho důvodu se nejprve zaměříme na technologie těchto společností (Adobe – Adobe Flex, Microsoft – Silverlight, Oracle – JavaFX, Google – Google Web Toolkit). Abychom v analýze neměly jenom technologie velkých společností, zaměříme si i na zajímavé technické řešení menší firmy. Představím technologii OpenLaszlo. Existuje spousta dalších technologií, které se zabývají tématikou oblasti diplomové práce (RIA). Cílem není čtenáře seznámit se všemi nástroji pro tvorbu Rich Internet Applications, které jsou na počátku svým vývojem a doposud se značně neprosadili na trhu RIA aplikací. Z toho důvodu srovnávám dle mého názoru pět nejznámějších technických řešení v rámci oblasti RIA. Výjimkou může být technologie OpenLaszlo, která je v očích českých vývojářů dosti neznámým pojmem. Najít české internetové stránky, které by se přímo zabývali touto technologií, je skoro nemožné. Proč tedy budu zmiňovat „nepopulární“ technické řešení? Domnívám se, že bychom byli značně ochuzeni tím, kdybych se soustředil pouze na nástroje společností s vysokým kapitálem. Pojem OpenLaszlo sice není u nás tolik rozšířený, ale naopak ve světě je technologie hojně

využívaná. Z tohoto důvodu jsem ji zařadil do seznamu srovnávaných technologií. Další výjimku tvoří koncept AJAX. Jak sem již uvedl v předcházející kapitole, pomocí kombinace známých technologií se dají vytvářet robustní internetové aplikace, ale v rámci své práce se soustředím na druhou možnost, tedy aplikace vytvořené pomocí vlastních prostředků a technologií. Navíc se v práci zmiňuji o technologii Google Web Toolkit, která poskytuje podporu vývojářů při tvorbě rozsáhlých, na AJAXu založených aplikací.

4.0.2 Analýza RIA technologií

Na začátku každé analýzy je důležité se seznámit se zkoumanými objekty. V našem případě se jedná o technologie. Každý nástroj krátkým úvodem představím, seznámím čtenáře s jeho vlastnosti, případně k čemu slouží nebo jak funguje. Dalším důležitým bodem je upozornit na hlavní skriptovací nebo značkovací jazyk, které dané technologie vyžívají při tvorbě bohatých internetových aplikací. K tomu, abychom mohli vytvářet aplikace danými technologiemi je zapotřebí mít kvalitní vývojové prostředí. Proto se u každého řešení zmíním i o nejpoužívanějších vývojových nástrojích pro danou technologii a jiných vlastností. Aby bylo možné nějakým způsobem srovnat (porovnat) technologie mezi sebou, bylo nutné vymezit několik relevantních hodnotících kritérií, které budou následně slovně popsány podle důležitých kvalitativních specifikací dané technologie. Technologie budu srovnávat z pěti širokých kritérií:

1. *Obecné vlastnosti* - V rámci tohoto kritéria se zaměřím především na historii technologie, zda se stále vyvíjí nové a aktualizují současné verze, možnosti technologie, základní architekturu (struktura klientské části a načtení do internetového prohlížeče), popř. budoucí vývoj, standardy atd.
2. *Vývojové prostředí* - Jak lze získat dané prostředí, za jakých finančních podmínek, instalace prostředí, množství jednotlivých vývojových nástrojů a jejich kvalita atd.
3. *Uživatelský pohled* - Obsahuje runtime prostředí, řešení na straně klienta, jaký je vzhled vytvořené aplikace, rychlost aplikace v rámci požadavků uživatele, osvojení uživatelů, mobilní podpora, uživatelské rozhraní, multiplatformovanost atd.
4. *Technologie z pohledu vývojáře* - Zaměření na množství psaného kódu, systémové požadavky, programování, komunitu vývojářů, podporu multimédií, dokumentaci technologie, otevřenost, SDK - vývojové nástroje, rychlost osvojení nástroje, údržbu aplikace, programovací jazyk a další.

5. *Reference* - Vymezuje nejznámější aplikace vytvořené pomocí dané technologie, dostupnost informací, reklama, pozice firmy na trhu atd.

Uvedený přehled zaměření v rámci daného kritéria určitě není kompletní, ale aspoň nastiňuje, jakým směrem se ubírá hodnocení konkrétního kritéria.

4.0.3 Metodika bodového ohodnocení jednotlivých technologií

Srovnávání technologií je provedeno bodovým ohodnocením jednotlivých kritérií. Jelikož dané kritéria nelze přesně změřit nebo exaktně srovnat, jedná se spíše o moje subjektivní názory. Každé ohodnocení se budu snažit podložit odpovídajícím odůvodněním, ať je zřejmé, proč byl přidělen právě takový počet bodů danému kritériu v rámci technického řešení. Metodou párového srovnávání stanovím důležitost jednotlivých kritérií. Stupnice vah je od 1 do 5. Čím je hodnota váhy vyšší, tím více se hodnocení promítne do konečného výsledku. Důležitost kritéria bude zvolena s ohledem na vývojáře, který se bude rozhodovat pro konkrétní řešení jeho problému a nebudou ho tolik zajímat dostupnost informací či historie. Máme tedy pět kritérií, které budou ohodnoceny na stupnici od 0 do 100. Veškeré varianty (technologie) budou ohodnoceny z hlediska užítka v rámci matice vážených užitečností, která bude popsána na konci této kapitoly.

4.0.4 Stanovení důležitosti kritérií

Ještě před samotným srovnáváním technologií si stanovíme důležitost jednotlivých kritérií metodou párového srovnávání a stanovíme jim váhy. Následující tabulka (tab. 4.0.4.1) ukazuje přehledně důležitost a srovnání kritérií.

Č.	Kritéria				
1.	Obecné vlastnosti				
		2.			
2.	Vývojové prostředí		1.		
		2.		4	
3.	Uživatelský pohled		4.		5.
		4.		2.	
4.	Techn. z pohledu vývojáře		3.		
		4.			
5.	Reference				

Tab. 4.0.4.1 - Párové srovnání

V následující tabulce je zobrazeno stanovení vah jednotlivým kritériím jako výsledek předchozí tabulky (tab. 4.0.4.1).

Č.	Kritéria	I.	II.	III.
		Výskyt	Pořadí	Váha
1.	Obecné vlastnosti	1x	3.	3
2.	Vývojové prostředí	3x	2.	4
3.	Uživatelský pohled	1x	3.	3
4.	Techn. z pohledu vývojáře	4x	1.	5
5.	Reference	1x	3.	3

Tab. 4.0.4.2 - Stanovení vah jednotlivých kritérií

Z výše uvedených výsledků jasně vyplývá, že nejdůležitějším kritériem pro výběr technologie je 4. - Technologie z pohledu vývojáře. Na druhém místě se umístilo kritérium 2. - Vývojové prostředí. Není žádným překvapením, že právě tyto dvě kritéria jsou pro vývojáře aplikací nejdůležitější, když vezmeme v potaz to, že žádný vývojář nezačne programovat aplikaci v technologii, která je postavena na programovacím jazyku, kterému nerozumí nebo v něm nikdy nevyvíjel žádnou aplikaci. Nezbytná je i volba vhodného a známého vývojového prostředí, s ním má vývojář už nějaké nemalé zkušenosti. Pro někoho může být překvapením, že zbylé tři kritéria se vyskytují ve srovnání vždy jednou. Je to dáno tím, že jsem se velice těžce rozhodoval, které kritérium upřednostním před druhým z pohledu vývojáře. Všechny tři kritéria jsou rovnovážná svou důležitostí a vývojář by neměl dělat mezi nimi rozdíl ve výběru technologie pro své řešení. Jak jsou podstatné vlastnosti zvolené technologie, tak nesmí podcenit při svém rozhodování ani systémové požadavky, runtime prostředí, mobilní podporu nebo budoucí vývoj technologie významnou společností na trhu.

4.1 Adobe Flex

Adobe Flex představuje souhrn technologií pro vývoj a nasazení bohatých internetových aplikací založených na platformě Adobe Flash. Aplikace Flex umožňuje vytvářet webové a mobilní aplikace, které sdílejí stejný kódový základ, čímž se snižuje doba a náklady na tvorbu aplikace a zároveň se prodlužují intervaly údržby. Zatímco aplikace Flex umožňuje vytvářet sestavy pouze pomocí volně dostupného prostředí Flex SDK, software Adobe® Flash® Builder™ umožňuje zrychlení vývoje prostřednictvím funkcí, jako jsou inteligentní úprava kódu, ladění v krocích, paměťové a výkonnostní profily a vizuální návrhy. Ačkoli je schopnost běhu Flex aplikací ve Flash Playeru jednou z jeho největších výhod, v komunitě webových vývojářů slovo Flash většinou vzbuzuje spíš odpor než sympatie. Bude proto dobré si vyjasnit, jak je to se vztahem Flexu k nástroji Flash. Když to řeknu jednoduše,

tak technologie Flex a Flash nemají společného prakticky nic kromě běhového prostředí (Adobe Flash Player). Uvedeme si pár rozdílů:

- Jak jsem již uvedl dříve, tak technologie Flash slouží k tvorbě animací, bannerů a dalších grafických prvků. Mezi základní návrh patří časová osa, klíčové snímky, přechody mezi nimi. Oproti Flashi je Flex programátorskou záležitostí, žádná časová osa zde neexistuje, vývojář komponuje uživatelské rozhraní ze znovupoužitelných prvků, celá aplikace je programovaná v moderním objektově orientovaném jazyce atd.
- Zdrojový kód Flashe je uložen v binárním souboru, který má koncovku „.fla“. K jeho editaci je zapotřebí Flash IDE. Naproti tomu zdrojové kódy Flexu jsou obyčejné textové soubory. K jejich editaci poslouží libovolný textový editor.
- Vývojovým prostředím Flashe je „Flash authoring tool“ (Flash MX, Flash Professional 8, Flash Professional CS5 a spousta dalších verzí). Vývojovým prostředím pro Flex je Flash Builder 4.5 nebo nějaký textový editor spolu s kompilátory příkazové řádky.

Společnost Adobe postavila technologii Flex nad Flash Playerem, tím dala možnost aplikacím běžet skoro ve všech moderních internetových prohlížečích světa. Navíc opustili od všeho, co se vývojářům nelíbilo na technologii Flash a jako náhradu nabídli framework, který je orientovaný na vývojáře. Flash má samozřejmě stále své dobré místo využití a pro společnost Adobe není problém spojit flashové animace s Flexem. [19]

4.1.1 Historie

Macromedia Flex 1.0, 1.5

První vydání je z března roku 2004 od firmy Macromedia (dnes už Adobe) jménem Flex Data Services (Macromedia Flex 1.0). Další verze Flex 1.5 vyšla ve stejném roce jako první verze. Aplikace byly psány objektově orientovaným jazykem ActionScript ve verzi 2.0. Pro vývoj aplikací byla k dispozici volně stažitelné zmíněné vývojové prostředí Flex Builder IDE a dostupná sada vývojových nástrojů nebo knihoven (SDK). Nástroj Flex Data Services zajistil snadnou manipulaci s daty. Jedná se o integrovaný nástroj vybudovaný na platformě Java 2 Enterprise Edition.

Adobe Flex 2.0 a 2.0

Adobe významně změnilo licenční model produktové řady Flex s vydáním Flex 2. Tato verze vyšla v roce 2006. Jádro Flex 2 SDK se skládalo z příkazového řádku, kompilátoru, kompletní knihovnou tříd prvků uživatelského rozhraní a vše bylo k dispozici k bezplatnému stažení. Společnost Adobe spolu s novou verzí Flexu vydala i nové vývojové prostředí Adobe Flex Builder 2. Flex Builder 2 byl postaven na platformě Eclipse a existoval ve dvou verzích (Standard a Professional). Liší se od sebe tím, že verze Professional obsahuje navíc knihovnu „Flex Charting Components“, která pomocí grafů umožňuje snadnou prezentaci dat.

Novinkou oproti předchozím verzím byla synchronizace dat, data push (přenos dat řízený odesílatelem), publish-subscribe a automatizované testování. Metoda Publish/Subscribe představuje zcela odlišný přístup k zasílání zpráv mezi systémy. Namísto klasického zasílání na základě adresy či jména příjemce probíhá doručování zpráv podle jejich obsahu. [7] I programovací jazyk ActionScript se dočkal nové verze (3.0), který je již v plném souladu se standardem ECMAScript a umožňuje definovat aplikační logiku na straně klienta. Samotné aplikace běžely v nově použitém běhovém prostředí Flash Playeru 9. Až další verze 2.01 přinesla podporu CSS stylů a přidávání modulů.

Adobe Flex 3

V roce 2007 společnost Adobe oficiálně oznámila svůj záměr vydat technologii Adobe Flex 3 SDK. Než spatřila světlo světa finální verze, vyšlo několik beta verzí, které pomohli k vydání konečného produktu v roce 2008. Oproti předcházejícím verzím přichází s několika zásadními vylepšeními: [4]

- *Knihovna tříd a aplikační služby* - Obsahuje předem sestavenou knihovnu tříd a aplikačních služeb, která pomáhá vývojářům sestavovat a vytvářet aplikace RIA. Mezi tyto služby patří navázání dat, řízení přetažení, zobrazovací systém, který řídí rozvržení uživatelského rozhraní, systém stylů, který řídí vzhled komponent rozhraní, a efekty a systém animací, který řídí pohyb a přechody.
- *Komponenty* - Knihovna komponent nabízí všechny ovládací prvky uživatelského rozhraní, které vývojáři potřebují, od jednoduchých tlačítek, zaškrťovacích polí a přepínačích tlačítek po komplexní datové mřížky, kombinovaná pole s nabídkami a editory RTF textu. Vývojář může použít poskytované kontejnery ke snadnému

návrhu složitých adaptivních rozvržení a použít (nebo modifikovat) vizuálně skvělé skiny k dosažení ideálního vzhledu a dojmu.

- *Integrace s Adobe AIR* - Runtime prostředí Adobe AIR™ rozšiřuje webové aplikace na plochu počítače nezávisle na prohlížeči. To vytváří nové příležitosti pro poutavější online/offline aplikace s vyšším výkonem. Framework Flex poskytuje nativní podporu pro nová rozhraní API AIR a program Flex Builder 3 obsahuje všechny potřebné nástroje k vytváření, ladění, sbalení a podepsání aplikací vytvořených na základě Adobe AIR.
- *Zmenšení velikosti souboru SWF* - Můžete podstatně zmenšit velikost souborů SWF svých aplikací Flex. Navíc k optimalizaci byte kódu v kompilátoru Adobe Flex 3 nová funkce v softwaru Adobe Flash® Player dovoluje trvalé uložení frameworku do vyrovnávací paměti. Framework Flex se uloží do vyrovnávací paměti při prvním použití libovolné aplikace Flex a je připraven k opakovanému použití s ostatními aplikacemi Flex bez ohledu na to, ze které domény pocházejí. Tato vyrovnávací paměť je nezávislá na vyrovnávací paměti prohlížeče, takže když byl framework Flex již jednou stažen, bude nadále k dispozici pro všechny aplikace Flex.
- *Práce se vzdálenými objekty a posílání zpráv ve Flexu* - Práce se vzdálenými objekty (Remoting) a posílání zpráv (Messaging) ve Flexu jsou nyní dostupné pro všechny vývojáře přes open source projekt BlazeDS. Vývojáři mohou nyní začít používat tyto výkonné funkce pro integraci se serverem Java™ zdarma, a pak přejít na Adobe LiveCycle® Data Services, Community Edition pro certifikované verze a podporu, nebo upgradovat na plnou verzi LiveCycle Data Services ES, poskytující kompletní serverové řešení.



Obr. 4.1.1 – Adobe Flex 3

Adobe Flex 4 a současnost

V červenci roku 2008 oznámila společnost Adobe vývoj nové verze Adobe Flex s kódovým označením Gumbo, který bude podporovat Adobe Flash Player 10. Přináší tři základní novinky – Framework Evolution, Developer Productivity, Design Mind. [17]

- *Design in Mind* - Poskytuje framework, umožňující plynulou spolupráci mezi návrhářem a vývojářem. Design in Mind přichází s novými komponentami, které vyhovují modelu pro snadnou úpravu vzhledu, nový model rozvržení komponent, novou syntaxi, nové efekty, rozhraní grafických tříd, novou verzi jazyka MXML a FXG₂₉. Umožňuje převést grafické objekty vytvořené v Adobe Flash do prostředí Flex.
- *Developer Productivity* - Tato vlastnost se snaží poskytnout zákazníkům návyky, na které jsou zvyklí v ostatních programovacích jazycích a dosáhnout co největší kompatibility se stávajícími normami. Tato snaha přináší výkonný kompilátor, obousměrnou vazbu komponent, zdokonalení využití CSS a automatickou podporu v Adobe AIR.
- *Framework Evolution* - Přináší podporu obousměrného textu a rozložení komponent. Přináší nové video komponenty a využívá nové možnosti Adobe Flash Playeru 10.

V současnosti je nejnovější verzí Adobe Flex 4.5, která přichází např. s novými Spark komponentami pro oddělení dat od chování, vylepšenou podporou pro globalizované aplikace, optimalizací velikosti aplikace, s nástrojem „Launchpad“. Launchpad slouží ke snadnému vytváření aplikací v AIR / Flex. V jednoduchém průvodci si vyberete základní parametry své aplikace a určíte, jaké komponenty chcete a budete používat a Launchpad vygeneruje potřebné soubory a kódy, které následně stačí jen naimportovat do Flash Builderu. [17]

4.1.2 Deklarativní jazyk pro popis uživatelského rozhraní - MXML

Pro vývoj ve Flexu se používají dva jazyky. Prvním z nich, který si uvedeme, je MXML. Primárně slouží pro tvorbu grafického rozhraní aplikace. MXML je jazyk postavený na XML, který je intuitivní a snadno čitelný stroji i lidmi. Proto předpokládám, že nikomu nebude dělat potíže. Soubory deklarativního jazyka mají koncovku „.xml“. Jazyk Macromedia eXtensible Markup Language se částečně podobá jazyku HyperText Markup Language, protože obsahuje značky pro definici uživatelského rozhraní. Pracuje se s ním

podobně jako s HTML, ale podstatný rozdíl je v tom, že MXML je strukturovanější a zahrnuje o mnoho víc značek.

Ukázka MXML:

```
<?xml version="1.0" encoding="utf-8"?>

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">

    <mx:Label x="200" y="250" text="Hello world" color="#000000" fontWeight="bold"/>

</mx:Application>
```

První řádek je povinný pro jakýkoliv XML soubor a sděluje nám, že jde o verzi 1.0 a znakovou sadou UTF-8. Druhý řádek obsahuje kořenovou značku naší aplikace. No a konečně jsme se dostali ke třetímu řádku. Tady voláme komponentu Label, která má zde jediný účel. Vypsát text „Hello world“ černou barvou (color=„#000000“) a normální šířkou (fontWeight=„normal“) na vzdálenosti 200px od pravého okraje (x=„200“) a 250px od horního okraje aplikace (y=„250“).

4.1.3 ActionScript

Druhým plnohodnotným programovacím jazykem je ActionScript 3. Jako většina dnes používaných jazyků vychází ze syntaxe jazyka C+. Nejvíce nadchne milovníky Javy, protože jsou si velice podobné. Jazyk je to silně typový a objektově orientovaný. Samozřejmě můžeme ActionScript použít i pro programování vzhledu (dokonce se do ActionScriptu v jedné z fází kompilace přepisuje MXML, ale o tom později), nicméně se v praxi používá spíše pro programování samotného fungování aplikace, tedy pro aplikační logiku. Zdrojový kód ActionScriptu můžeme zapisovat třemi postupy. První postup je zapisovat kód přímo do daného elementu pomocí složených závorek „{}“ a tečkové notace. Tímto způsobem lze zapisovat jen velmi jednoduché příkazy. Další postup zápisu kódu je přímo do souboru Macromedia eXtensible Markup Language s použitím vyhrazené značky „Script“. Poslední způsob je vytvoření samostatného souboru s koncovkou .as, do kterého je umístěn zdrojový kód ActionScriptu. Následně je nutné tento soubor připojit k MXML souboru pomocí elementu „Script“ a jeho atributu „source“. Následuje ukázka, která dělá to samé, co předchozí ukázka v předešlé podkapitole:

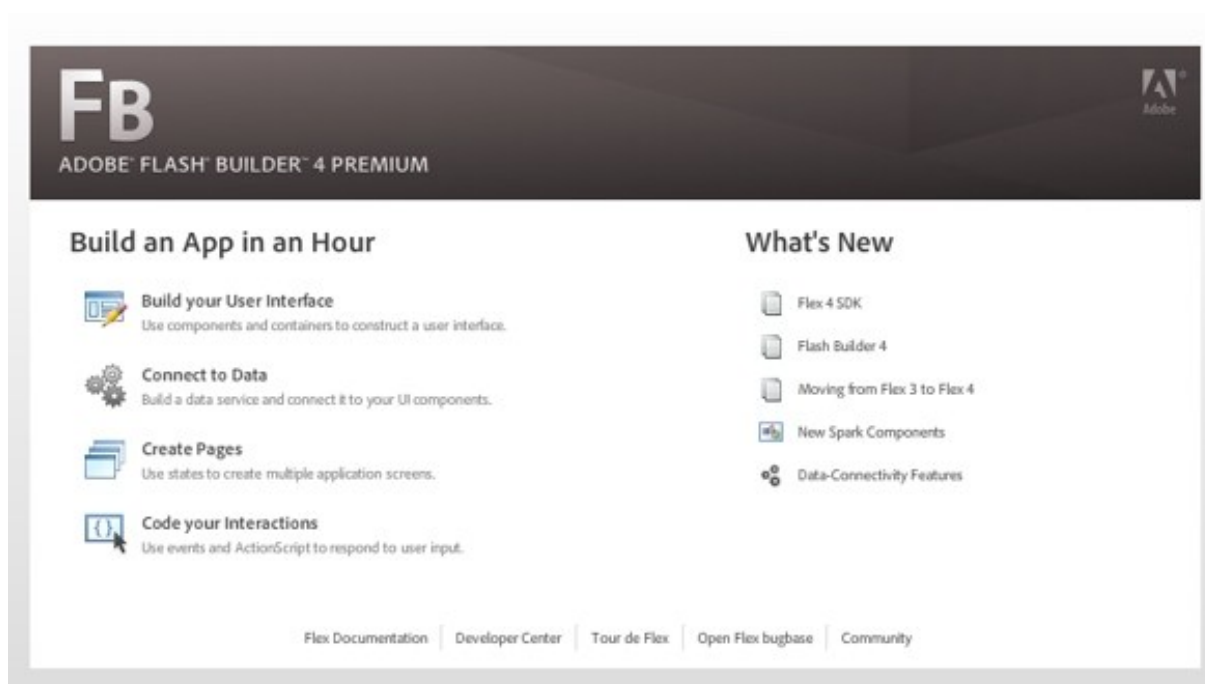
```
var text:Label = new Label();  
  
text.text = "Hello world";text.x = 250;  
  
text.y = 200;  
  
text.setStyle("color", "#000000");  
  
text.setStyle("fontWeight", "normal");
```

ActionScript byl navržen pro znázornění nového programového modelu, srozumitelného všem vývojářům se základními znalostmi objektově orientovaného programování. Od tohoto jazyka je požadováno dosažení čtyř hlavních cílových oblastí [20]:

1. *Bezpečnost* – dosažení typového zabezpečení pro tvorbu jednoznačného a lehce udržitelného kódu.
2. *Jednoduchost* – dosažení snadného pochopení programu bez nutnosti neustálého nahlížení do dokumentace.
3. *Výkon* – dosažení možnosti vytvořit komplexní účinný program
4. *Kompatibilita* – dosažení zpětné a přední kompatibility a dodržení standardů

4.1.4 Vývojové prostředí – Adobe Flash Builder

Adobe Flash Builder 4.5 je vysoce produktivní, open source a bezplatné prostředí pro tvorbu působivých webových aplikací, které lze konzistentně prohlížet v různých počítačích, prohlížečích a operačních systémech pomocí programů Adobe Flash Player a Adobe AIR. Aplikace Flex jde vytvářet pomocí samotného prostředí Flex, avšak software Flash Builder (dříve Adobe Flex Builder) umožňuje vývoj urychlit pomocí různých funkcí, např. interaktivního ladění s krokováním a vizuálním návrhem rozložení uživatelského rozhraní nebo inteligentního kódování. Adobe Flash Builder je dostupný ve dvou verzích Standart a Premium. Navíc je k dispozici v podobě SDK pro Eclipse, které je zdarma. Pouze verze Premium zabezpečí dobrý základ pro vývoj flashových aplikací a to především díky přítomnosti nástrojů pro ladění a sledování chodu aplikace. Dobrou zprávou je, že součástí volného Flex 4 SDK jsou i komponenty grafů a Advanced Datagrid, za které jsme si museli doplácet v dřívějších verzích.



Obr. 4.1.4 - Adobe Flash Builder 4 Premium

4.1.5 Runtime prostředí

Webové aplikace vytvořené pomocí technologie potřebují pro svůj běh runtime prostředí (běhové prostředí). Společnost Adobe nabízí tři běhová prostředí:

- Adobe Flash Player
- Adobe Flash Lite
- Adobe AIR

Adobe Flash Player byl původně navržen pro přehrávání dvojrozměrné vektorové grafiky, čímž dokázal nabídnout zajímavé prezentace. S rostoucími požadavky uživatelů Internetu se z něj stal nejrozšířenější přehrávač obsahu internetových stránek. Dnes si jen málokdo dokáže představit, že by se bez něj mohl obejít, protože byste si nemohli spouštět videa na youtube či streamu. V současnosti je multiplatformní aplikační běhové prostředí založené na prohlížeči a určené pro prohlížení obsahově bohatých aplikací, obsahu, videa na různých obrazovkách a prohlížečích. Díky aplikaci Flash Player 10.2 získáte skvělé HD video, rychlejší vykreslování grafiky a vysoký výkon na mobilních zařízeních. [21] Adobe Flash Lite je odlehčenou verzí předešlého runtime prostředí. Tato verze je určena pro mobilní zařízení a další přenosné elektronické zařízení. Umožňuje uživatelům těchto zařízení zobrazení multimediálního obsahu a aplikací vyvinutých pomocí Adobe Flash nástroje, které byly dříve k dispozici pouze na osobních počítačích. Adobe AIR (Adobe Integrated Runtime)

je relativně mladý projekt firmy Adobe, který aktivně vyvíjí a zlepšuje již několik let. Jedná se o zcela nové runtime prostředí pro aplikace, které umožní těm, kteří mají znalosti s programováním webových aplikací a stránek vytvořit desktopovou aplikaci. Vše s použitím současných znalostí pro vývoj webu jako je: Ajax, Adobe Flex, Adobe Flash nebo HTML. Aplikace, které vytvoříme pro AIR, mají nesmírnou výhodu v tom, že běhové prostředí AIR je dostupné pro téměř všechny nejpoužívanější operační systémy, konkrétně Linux, Windows a Macintosh. Z toho vyplývá, že svou oblíbenou aplikaci spustíte na každém z těchto systémů. Instalace je velice snadná, stejně jako přidávání nových aplikací.

4.1.6 Kritéria hodnocení

Obecné vlastnosti - Adobe Flex se vyvíjí dlouhou dobu, ale stále má krátkou historii. Od svého vzniku prošla řadou změn, oprav a byla vydána v mnoho verzích. Oproti konkurenci má tu výhodu, že Adobe s vývojem technologie pro tvorbu RIA začalo jako první. Z toho vyplývá mírný náskok v oblasti robustních internetových aplikací. Možnosti technologie jsou velmi bohaté. Základní architektura je vázaná na Flash přehrávač a pro překlad do flashové podoby je potřeba kompilátoru a odpovídajících knihoven. V HTML stránce je uveden odkaz ve funkci, kterou načte JavaScript. Při načtení stránky je na serveru nalezen příslušný swf soubor. Tento soubor je natažen do Flash přehrávače, který jej následně spuštěn. Budoucnost nástroje Adobe Flex není ohrožena, protože společnost Adobe nezahálí a stále se snaží vyvíjet nové verze a upgradu ke stávající verzi. Podporuje internacionalizaci, formáty XML, JSON, SOAP a standard ECMAScript.

Díky tomu, že technologie Flex se vyvíjela dříve než u konkurence, tak má náskok v oblasti vývoje robustních internetových aplikací. Navíc disponuje kvalitním a nejznámějším runtime prostředím. Uděluji 100 bodů.

Vývojové prostředí - Plnohodnotný a pohodlný vývoj aplikací technologií Adobe Flex je umožněn ve vývojovém prostředí Adobe Flash 4. Vývojový nástroj je komerční, tudíž si ho každý nemůže dovolit. Adobe poskytuje 30 odzkoušení zdarma, poté je nutné produkt zakoupit. Adobe poskytuje pro studenty zdarma verzi, která je okleštěná o několik funkcí. Navíc funguje pouze na platformách Windows, Mac OS a má náročné systémové požadavky oproti jiným prostředím. Existují dvě verze: Standard a Professional. Obsahuje standardní funkce, jako jsou např. automatické testování, nástroje pro ladění, inteligentní dosazování kódu a velice zajímavé prostředí pro vývoj uživatelského rozhraní. Navíc produkt spolupracuje s ostatními produkty společnosti Adobe (např. S pomocí Adobe AIR je možné

výslednou aplikaci provozovat přímo na desktopu počítače). Instalace prostředí není složitá a rychlost záleží na hardwaru počítače.

Flex disponuje kvalitním vývojovým nástrojem, bohužel velkou nevýhodou je „proprietárnost“ a možnost použití jednoho vývojového prostředí. Přiděluji 85 bodů.

Uživatelský pohled - Technologie Adobe Flex má nesmírnou výhodu v tom, že využívá runtime prostředí Adobe Flash, které je skoro všude dostupné (99% uživatelů internetu). Jeho silná stránka může být i hlavní nevýhodou, protože ne na všech platformách či zařízeních může běžet stoprocentně (zejména na operačním systému Linux). Podporuje množství komponent, které může tvůrce jednoduše vzít a zakomponovat do své vytvářené aplikace a tím jsou možnosti technologie opravdu veliké z pohledu vývojáře. Instalace runtime prostředí není obtížná a tím pádem i spuštění aplikací je jednoduché. Malou nevýhodou může být načtení většího objemu dat před spuštěním. Adobe Flex je velice multiplatformní, může běžet na platformách Windows 7 (32 bit), Windows Vista (32 bit), Windows XP Professional s SP2 nebo SP3, Mac OS X v10.5.6–10.6 (pouze Intel), Ubuntu 9.10, Fedora Core 12. Systémové požadavky na pracovní stanici jsou běžné vzhledem k vývoji hardwaru. Nasazení aplikace na server nenese sebou žádné omezení, odvíjí se pouze od specifikací internetového serveru. S podporou mobilních zařízení zatím nástroj Flex zaostává oproti konkurenci. Výsledný vzhled aplikace je působivý díky podpoře kaskádových stylů, grafických komponent a dalších.

Technologie těží hlavně z rozšířeného Adobe Flash Playeru, který běží skoro na všech platformách. Nevyvíjí se mobilní podpora. Dávám 95 bodů.

Technologie z pohledu vývojáře - Vývoj aplikací je pomocí dvou jazyků: MXML a ActionScript. Používání jazyka MXML je jednoduché a programátor zvyklý na Javu si rychle osvojí ActionScript. Podrobnosti o jednotlivých jazycích jsem již zmínil. Při kompilaci se kompletní MXML kód převede do ActionScriptu a ten následně do SWF bytekódu. Adobe Flex je open source pod Mozilla Public License. Uvedené runtime a vývojové prostředí jsou komerční softwary. Vývojový nástroj obsahuje kompletní dokumentaci, runtime prostředí, sadu knihoven, nástroje pro vývoj aplikace a ukázkové příklady. Nástroj podporuje audio formát MP3. Nevýhodou je potřeba převádět video formáty do FLV (Flash Video). Společnost Adobe poskytuje pro Flex rozsáhlou dokumentaci k rozhraní pro programování aplikací, soupis UI komponent, komunikaci se serverem, zpracování událostí, třídy pro správu barev a fontů, tutoriály, ukázkové příklady demo aplikací, registrované diskuzní fórum, popis

a syntaxi jazyka v příručce programátora. V tomto ohledu se Adobe nedá nic vytknout. Knihovny obsahují více než 100 komponent, podporů grafů, animací a stylů. Pro získání pokročilé úrovně programování v technologii Adobe Flex je potřeba o hodně více času než třeba u nástroje GWT, protože technologie je velice široká svým pojetím, ale čas nám zkrátí právě dokumentace, tutoriály a další. Ve Flexu se špatně programují aplikace podobné malování (grafický editor). Předností nástroje je bezproblémová podpora data bindingu. Pomocí vývojového prostředí Flash Builder je vývoj aplikací velmi rychlý, naopak každé zkompileování do formátu SWF rychlost vývoje zpomaluje. Množství kódu potřebného k napsání aplikace je malé, protože Flex je nutné psát pouze kód, který má funkční smysl ve spojitosti s aplikací. Oddělení MXML kódu a funkčních skriptů přispívá k přehlednosti zdrojového kódu. Další výhodou je dobrá čitelnost zdrojového kódu, není složitý a snadno rozšiřitelný. Díky překladu kódu do formátu SWF není zpřístupněn klientovi. Technologie Flex se bezesporu spíše hodí ke tvorbě desktopových aplikací než k programování klasických dynamických webů, protože není způsobilý pro práci s HTML. Lepší variantou je přímé vložení aplikace do HTML stránky.

Z uvedených informací je patrné, že pokud vývojář vsadí na Adobe Flex, určitě nebude zklamán. Přiděluji 95 bodů.

Reference - Nesmírnou výhodou tohoto nástroje je podpora významné společnosti na trhu - Adobe. Existuje relativně (krátká historie technologie) velká základna Flex vývojářů, kteří vyvinuli opravdu velké množství známým aplikací např. SpatialKey - systém GIS, Fox Racing - internetový portál, Nasdaq Market Replay - simulace trhu pro finanční odborníky, Times Reader 2.0 - aplikace ke čtení známých novin v USA a spousta dalších aplikací. Na internetu a v knížkách se dá najít mnoho informací o technologii Flex. V tomto určitě nezaostává za konkurencí.

Společnost Adobe nemá problém s distribucí své technologie, která je velice známá, patří k nejlepším technologiím na trhu v oblasti RIA a vzniklo hodně projektů postavených na tomto nástroji. Dávám 95 bodů.

4.2 Microsoft Silverlight

Společnost Microsoft se poslední dobou stále více zajímá o internetové technologie. Jednalo se především o technologii ASP.NET, jakožto oblíbený, rozšířený a silný nástroj pro tvorbu dynamických webových stránek a aplikací. [5] Silverlight je spojení existujících

technologií za účelem získat z nich ty prvky, které jsou pro RIA aplikace vhodné. Podobné to bylo i v případě Adobe Flex, který vycházel z technologie Adobe Flash a přidával k němu vhodné prostředky pro tvorbu RIA aplikací. Microsoft představil ve Windows Vista nové technologie, jako jsou Windows Presentation Foundation nebo Windows Communication Foundation. Díky prvně zmíněné se na trhu objevila nová technologie – Microsoft Silverlight ze které funkčně vychází. Silverlight je Microsoftem prezentovaná jako nejbližším konkurentem pro technologii Adobe Flash. Hlavní cílem je dosažení předních míst v rámci technologií na Internetu vedle Adobe Flash či AJAXU. Silverlight nabízí kombinaci vektorové grafiky, animaci textu, bitmapové grafiky, umožňuje přehrávat HD videa a především vyniká možností libovolné volby použitého programovacího jazyka .NET od verze 2.0 a výš. V první verzi (1.0) byla funkcionality zajištěna jedině JavaScriptem. K zobrazení aplikací, které jsou vytvořeny pomocí technologie Silverlight je potřeba mít nainstalovanou komponentu (plugin), který obstarává jeho funkčnost ve většině moderních webových prohlížečů (Internet Explorer, Firefox, Safari, Opera, Chrome) na platformách Windows a Mac OS X. Nebo mohou běžet v režimu „out of browser“, tedy ve vlastním okně mimo prohlížeč. Vydatně se pracuje i na pluginech do mobilních zařízení (Windows Mobile, Symbian) a existuje také volně šiřitelná implementace pro unixové operační systémy s názvem Moonlight vyvíjena společností Novell. Výhodou technologie je schopnost propojení Silverlight aplikace a webovou stránkou, na které je umístěna. Pomocí ovládacích prvků aplikace je možné měnit obsah HTML stránky a naopak. Silverlight aplikace jsou nejčastěji vyvíjené ve vývojovém prostředí MS Visual Studio 2010 ve spojení s .NET Framework 4. Aplikace se programují podobně jako v případě Adobe Flexu, základem je značkový jazyk, který definuje prvky uživatelského rozhraní, jejich reakce na různé události a provázání prvků s proměnnými. Účelem Silverlightu nikdy nebylo a není na webu plně nahradit HTML. Jedním z nejlepších a největších projektů postavených na Silverlightu je „WorldWide Telescope“, jež v sobě ukrývá největší interaktivní mapu hvězdné oblohy.

4.2.1 Historie

Microsoft Silverlight 1.0

Společnost Microsoft představila v září 2007 první verzi Silverlightu s kódovým označením Windows Presentation Foundation / Everywhere (WPF/E). Technologie byla podporována pouze v prohlížeči Microsoftu – Internet Explorer. Funkcionality byla vytvořena pomocí skriptovacího jazyku JavaScript, kdežto návrh aplikace se prováděl pomocí značkovacího jazyku XAML. Aplikace byly spouštěny vyvoláním ovládacího prvku

Silverlight, který následně načetl soubor XAML (Extensible Application Markup Language, čteno „zaml“). Přípona souboru se shoduje se základem technologie, kterou se stal značkovací jazyk XAML, který je založen na formátu XML. Tento jazyk slouží pro deklarativní popis prezentačního rozhraní aplikace dodnes. Pro obsluhu událostí deklarovaných prvků a vůbec definici aplikační logiky byl zvolen jazyk JavaScript. Společně s první verzí byla zároveň představena také alfa verze 1.1, která byla ve finální fázi přechíslována na Silverlight 2.0. [22] Technologie podporovala práci s daty ve formátech XML a JSON. Dokázal zpracovávat vstupy z klávesnice, myši a jiných polohových zařízení. Ovládal manipulaci s vektorovou grafikou, tvorbu animace, vykreslování bitmapových obrázků, přehrávání multimédií (formáty MP3, WMV7, WMV8, WMV9/VC1, WMA).

Microsoft Silverlight 2.0

Podstatná změna ve druhé verzi se odehrála v oblasti aplikační logiky. JavaScript na pozadí byl doplněn jazyky platformy .NET, kterými jsou například populární C# nebo Visual Basic. Použitím těchto jazyků tak bylo docíleno přístupu k takřka všem knihovnám .NET Frameworku a byla tak rozšířena funkcionality technologie. Společně s novými jazyky se změnil i způsob sestavení aplikace samotné. Deklarativní soubor XAML je společně s kódem na pozadí zkompileován do .NET assembly, která je zkomprimována pomocí ZIPu a uložena do souboru XAP. Avšak i přesto lze z řízeného kódu Silverlightu přistupovat k celé struktuře stránky a určitým prvkům jazyka HTML (tedy k celému DOM – Document Object Model) díky speciálním wrapperům, které komunikaci zajistí. Díky použitým knihovnám přibyla rovněž podpora podmnžiny technologie WPF, která obsahuje nástroje pro kreslení, podporu médií a objekty WPF pro animaci. Dále přibyla podpora vizuálních prvků (TextBox, CheckBox, Slider...), manažerů rozložení (Grid, StackPanel...) a také prvky pro práci s daty (DataGrid, ListBox...). Navíc jsou všechny tyto prvky skinovatelné díky použitým šablonám. S druhou verzí Silverlightu se také vylepšila podpora síťové komunikace. Podporovány jsou jak webové služby postavené na XML, tak i Windows Communication Foundation (WCF) a datové služby ADO.NET. [22]

Microsoft Silverlight 3.0

Další verze na sebe nenechala dlouho čekat a byla představena již v červenci roku 2009. Tato verze se stala zlomovou ve smyslu razantního zmenšení rozdílu mezi webovou a desktopovou aplikací. Od této doby již mohou Silverlight aplikace běžet i mimo prohlížeč v takzvaném režimu „Out of Browser (OOB)“. Uživatel si může aplikaci nainstalovat do

lokálního operačního systému, vytvořit si zástupce na pracovní ploše a v podstatě nepozná rozdíl od klasické aplikace. Taková aplikace si ovšem s sebou nese stejná omezení jako by běžela v prohlížeči. Nachází se v tzv. „sandboxu“, což je izolovaný prostor, který slouží k odstínění aplikace od přímého přístupu k systému. Je tak zajištěna ochrana proti škodlivým aplikacím. Vylepšení se dočkala podpora vázání dat (databinding) na úrovni XAML kódu a také i podpora pro business objekty. Důležitou novinkou je i podpora SEO, tj. podpora indexování pro internetové vyhledávače, a také tzv. Deep Linking, což umožňuje přechod do určitého stavu aplikace pomocí parametrů hypertextového odkazu. Ve třetí verzi se objevují i prvky pro snazší provádění animací, resp. pro jejich nelineární běh. Nyní lze pomocí tzv. Animation Easing například simulovat odrážející se předmět na úrovni XAMLu bez nutnosti programového zásahu do kódu na pozadí. Přibyla také podpora streamování videa ve vysokém rozlišení a funkce Smooth Streaming, která na základě rychlosti internetového připojení přizpůsobí kvalitu videa tak, aby přehrávání bylo plynulé. U videa ve vysokém rozlišení lze nově využít akceleraci grafické karty a ulehčit tak práci procesoru, který může zatím zpracovávat aplikační logiku aplikace. [22]

Microsoft Silverlight 4.0 a současnost

Krátce po uvedení komerční verze 3.0 byla představena první beta verze čtvrté generace. Mezi nejvýznamnější novinky této verze lze označit podporu webové kamery, mikrofonu a multicast streaming. Tvůrci aplikací pro podnikovou sféru určitě ocení nově zabudovaný Rich text editor (editor nabízí rozšířené formátování textu) a podporu schránky, což umožňuje například kopírovat záznam z prvku DataGrid do tabulkového procesoru Microsoft Office Excel apod. Své využití určitě najde i funkce drag&drop, funkce pravého tlačítka myši a schopnost vložení HTML stránky do aplikace. Objevuje se také možnost spolupráce s poštovním klientem Outlook nebo s hardwarem (jako např. GPS modul) díky COM automation. [22] V současné době společnost Microsoft intenzivně vyvíjí verzi Silverlight 5.0, které bychom se měli dočkat v nejbližší době. Silverlight 5 přinese více než 40 výraznějších novinek, které většině vývojářů doteď chyběly. V souladu s odklonem od všestranného řešení bude většina změn zaměřena na využití pro multimediální a podnikové aplikace. Za zmínku stojí vylepšení v podobě nativní podpory 64-bitových systémů, lepší využití GPU pro renderování 2D a 3D obsahu a také nové funkce týkající se přehrávání multimediálního obsahu. Hardwarová akcelerace pro H.264 videa je určitě vítanou změnou, stejně jako funkce „TrickPlay“, které umožní přehrávání obsahu v různých rychlostech, a tím i rychlé přetáčení „tam a zpět“. Nová verze bude mít i lepší správu pro DRM ochrany nebo

také lepší možnosti pro dálková ovládání. Především pro podniková řešení se jistě uplatní tisk formátu (jazyka) Postscript a pro plynulejší animace byla vylepšena podpora Fluid UI.

4.2.2 Značkovací jazyk - eXtensible Application Markup Language

Psaní grafického rozhraní pomocí objektově orientované jazyka je velice pomalé a nepřehledné (těžkopádné). Značkovací jazyky (markup languages) jsou pro definování vzhledu ideálním řešením – rychle, stručně, přehledně a jednoduše můžete nadefinovat vlastnosti, umístění a spojitosti jednotlivých elementů. XAML slouží pro psaní uživatelského rozhraní. Původně zkratka znamenala eXtensible Avalon Markup Language. Avalon je prezentační prostředí pro nové Windows s kódovým označením Longhorn. Grafická prezentace pro Avalon je tvořena právě značkovacím jazykem XAML. [3] Výhodou tohoto jazyka je oddělení uživatelského rozhraní od aplikační logiky, která je tvořena kódem v jazyce C# nebo Visual Basic .NET. Jazyk XAML se vyznačuje tím, že je snadno persistentní – však se také jedná o formát XML. Hlavní funkcí XAML Parseru je převod zápisu ve formátu stromu XAML elementů do instancí objektů .NET třídy. V současné době existují čtyři verze XAMLu:

- WPF – Windows Presentation Foundation
- XPS – fixní dokumenty taktéž produkované v Office
- Windows Workflow Foundation (WF)
- Silverlight

Typicky je v XAMLu definován design, vazba na datové zdroje, zpracování a zachycení událostí. Všechno, co se dá naprogramovat pomocí značkovacího jazyku XAML, je možné napsat i pomocí standardních .NET jazyků (C#, VB.NET). Výhodou XAMLu je velká jednoduchost. Následující ukázkový kód vypisuje text „Hello World“.

Ukázka kódu:

```
<Canvas xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <TextBlock>Hello World</TextBlock>
</Canvas>
```

4.2.3 C# a Visual Basic .NET

C# je moderní, objektově orientovaný a typově bezpečný programovací jazyk vyvinutý firmou Microsoft zároveň s platformou .NET Framework. C# je dle návrhu jednoduchý programovací jazyk patřící do rodiny jazyků C. Snadno si jej proto oblíbí uživatelé jazyků C, C++, Java, JavaScript (ECMAScript). Pro studenty může být naopak jazykem prvním a usnadnit jim seznamování s ostatními jazyky z této rodiny. Charakteristiky:[6]

- Jazyk C# je čistě objektově orientovaný.
- Obsahuje nativní podporu komponentního programování.
- Podobně jako Java obsahuje pouze jednoduchou dědičnost s možností násobné implementace rozhraní.
- Vedle členských dat a metod přidává vlastnosti a události.
- Správa paměti je automatická. O korektní uvolňování zdrojů aplikace se stará garbage collector.
- Podporuje zpracování chyb pomocí výjimek.
- Zajišťuje typovou bezpečnost a podporuje řízení verzí.
- Podporuje atributové programování.
- Zajišťuje zpětnou kompatibilitu se stávajícím kódem jak na binární tak na zdrojové úrovni.

C# lze využít k tvorbě databázových programů, webových aplikací a stránek, formulářových aplikací ve Windows, softwaru pro mobilní zařízení (PDA a mobilní telefony), webových služeb atd. Celý vývoj programovacího jazyka prošel několika etapami a v současnosti existuje ve verzi 4.0. Visual Basic .NET je moderní programovací jazyk, ve kterém můžeme snadno a rychle vyvíjet programy pro systém Windows, aplikace pro Pocket PC a webové aplikace. Visual Basic prodělal dosti výrazný vývoj a díky tomu vzniklo mnoho verzí. Klasický Visual Basic (to jsou verze 1 až 6), byl velmi populární, ale měl mnoho nesrovnalostí, které jej brzdily. Mimo to vznikl ještě VBA - Visual Basic for Applications, který najdete např. v sadě Microsoft Office. V roce 2002 nastal prudký zvrat - již zastaralý jazyk Visual Basic 6, kterému pomalu začínal docházet v porovnání s ostatními jazyky dech, byl zmodernizován a vznikl Visual Basic .NET. Jedná se vlastně o novou generaci tohoto jazyka, která si získává velkou oblibu a dnes tento jazyk patří ke špičce. Ve Visual Basic .NET můžete naprogramovat téměř vše od her až po rozsáhlé databázové aplikace a jiné

projekty. Prakticky jediné, co v tomto jazyce napsat nemůžete, jsou ovladače hardwarových zařízení a programy, které musí běžet bez operačního systému s podporou .NET. To však ve většině případů nevadí a prakticky jediný běžně používaný jazyk, který toto umí, je C++. [25] Visual Basic .NET pracuje na rozhraní .NET, které výsledný program nezkompiluje přímo do strojového kódu, ale do jakéhosi mezikódu, kterému se říká CIL (Common Intermediate Language). Tento mezikód se zabalí do souboru „.EXE“ a distribuuje se na počítače k uživatelům v této podobě. Není to přímo strojový kód. Až v okamžiku, kdy se program spouští, zkompiluje se strojového kódu. Navíc se kompiluje jen to, co je z programu potřeba před prvním spuštěním. Říká se tomu JIT – Just In Time. Výhoda je ta, že se může zkompilovat do strojového kódu a použít instrukce, které ovládá procesor. Ten kód spouští. Celý program se zoptimalizuje pro daný počítač tak, aby běžel co nejrychleji. Navíc kompilace probíhá pouze jednou, zkompilované části programu se někde uloží a při dalším použití programu se pouze spustí. Program tak teoreticky může běžet rychleji, protože využívá i pokročilé instrukce nových procesorů. V praxi to platí jen někdy, protože program je zase zpomalován kompilací, která probíhá až za běhu, i tak je ale výkon dostatečný.

4.2.4 Vývojové prostředí – Expression Blend 4, Visual Studio 2010, Visual Web Developer 2010

Pro vývoj aplikací Silverlight nabízí společnost tři vývojové prostředí: Microsoft Expression Blend 4, Microsoft Visual Studio 2010, Microsoft Visual Web Developer 2010. Expression Blend není nic jiného než XAML WYSIWYG (What You See Is What You Get – „Co vidíš, to dostaneš“) editor. Umožňuje snadno vytvářet barevné přechody, animace, upravovat stávající kontroly, prostě vše, co vás může okolo návrhu GUI aplikace vůbec napadnout. Samozřejmě obsahuje i přímý editor XAML kódu, jen C# (nebo VB.NET) kód musíte upravovat externě, třeba ve Visual Studiu. Hlavní zbraní je bezesporu silná podpora pro tvorbu aplikací určených na komunikátory Windows Phone 7 Series. Podporuje Silverlight 4 i WPF 4 a přirozeně bezchybně spolupracuje s Visual Studií 2010. Visual Studio je integrované prostředí, které zjednodušuje tvorbu, ladění a nasazování aplikací. Výkonné nástroje pro návrh a možnost svést dohromady návrháře a vývojáře vám dovolí plně uplatnit tvůrčí schopnosti a přeměnit vize ve skutečnost. Psaní kódu je rychlejší, využijete své dosavadní dovednosti, prostředí si můžete přizpůsobit podle svých zvyklostí a hotové aplikace lze provozovat na různých platformách (včetně SharePoint a Cloud), jejichž počet stále narůstá. Integrovaná podpora vývoje řízeného testy (Test Driven Development) a nové ladicí nástroje pomáhají snadno a rychle odhalit a opravit chyby a zajišťují tak vysokou

kvalitu hotových aplikací. Visual Web Developer 2010 nabízí výkonné a moderní nástroje pro a komfortní vývoj dynamických webů. Podporuje snadnou aplikaci CSS stylů, tvorbu aplikací využívajících JavaScript a AJAX (vestavěná podpora ASP.NET AJAX, debugging) a databázových aplikací s využitím LINQ (Language Integrated Query) – nativní podpora přímého připojení k MS SQL. Nabízí řadu předpřipravených šablon webů, které dále můžete upravovat. Jaký je mezi těmito dvěma vývojovými programy rozdíl? Visual Studio 2010 je profesionální vývojové prostředí společnosti Microsoft pro vytváření webových aplikací, stejně jako desktopových (Windows) aplikací, služeb systému Windows, komponent, ovládacích prvků a dalších aplikací. Oproti tomu, Visual Web Developer poskytuje zjednodušené rozhraní, které se zaměřuje na poskytování nástrojů, které potřebujete pro vytváření webových aplikací. Proto je Visual Web Developer více kompaktní pro tvorbu webu a pro práci může být snazší než Visual Studio.

4.2.5 Runtime prostředí – zásuvný modul Silverlight plugin

Silverlight je plugin vytvořený společností Microsoft pro prohlížeče Windows Internet Explorer, Firefox a Safari pro zobrazení multimediálního obsahu. Zásuvný modul je postaven na technologii Windows Presentation Foundation (WPF) a .NET Framework. Instalace .NET Framework na klientském počítači není vyžadována. Oproti konkurenčnímu pluginu Adobe Flash není tolik rozšířený. Zásuvný modul Silverlightu má působivý seznam funkcí, některé jsou stejné jako ve Flashi, některé jsou zcela nové nebo dokonce revoluční. Do nich patří [26]:

- *Široká podpora prohlížečů* – Funguje na platformách Windows XP, Windows Vista, Windows 7, Mac OS, Linux (projekt Moonlight) a ve webových prohlížečích Internet Explorer, Firefox, Safari, Opera, Chrome.
- *Je odlehčený* – To znamená, že má příjemnou instalaci jako např. Adobe Flash a soubor má velice malou velikost, tudíž je rychle stáhnutelný.
- *Interakce s webovými službami* – Aplikace Silverlight mohou volat webové služby ASP.NET nebo WCF
- *Média* – Přehrává Windows Media Audio, Windows Media Video, MP3 Audio a VC-1.
- *CLR* - Zahrnuje zredukovanou verzi CLR, prakticky kompletní, s vyčerpávající sadou tříd jádra, svozem odpadků (garbage collector), s kompilátorem JIT, s podporou generiky atd.

- *Animace* - Animační model založený na čase, který umožňuje definovat, co se má dít, a jak dlouho to má trvat.
- *Dvourozměrné kreslení* - Poskytuje bohatý model pro dvourozměrné kreslení.

Technologie Silverlight může prezentovat obsah, který máme na webu, ale atraktivnější způsob s kvalitně vyšší úrovní prezentace je použití pluginu ve webovém prohlížeči spolu s nástrojem Silverlight.

4.2.6 Kritéria hodnocení

Obecné vlastnosti - I přes svůj docela krátký vývoj je technologie Silverlight velice ambiciózní projekt společnosti Microsoft. Technologická nástroj Silverlight využívá stávajících technologických prvků platformy .NET Framework. Z toho důvodu nebude problém si osvojit Silverlight pro zaběhlé vývojáře platformy .NET. Díky tomu jsou technologické možnosti tohoto nástroje opravdu na výborné úrovni. Jako u předchozí technologie se s budoucností Silverlightu u Microsoftu nemusíme obávat. Společnost stále vyvíjí nové verze, stávající aktualizuje, opravuje technické chyby. Architekturu tvoří jednak prvky platformy .NET a prezentační Framework. První část obsahuje komponenty pro CLR knihovny, síťové služby či sjednocení dat. Druhá část zahrnuje značkovací jazyk XAML, komponenty a služby pro zobrazení a zpracování událostí uživatelského rozhraní. Spuštění aplikací je podobné jako u Adobe Flex s tím rozdílem, že stahovat aplikaci můžeme na požádání. Podpora JSON, XML, WMA, WMV formátů. Silverlight předčí svou konkurenci hlavně funkcí streamování multimediálního obsahu. [34]

Ačkoliv je Silverlight mladá technologie, společnost Microsoft z ní za tu dobu udělala jednu z nejlepších, ale stále za Adobe v něčem zaostává. Uděluji 95 bodů.

Vývojové prostředí - Pro vývoj aplikací pomocí technologie Silverlight existují tři plnohodnotné vývojové nástroje (Microsoft Expression Blend 4, Microsoft Visual Studio 2010, Microsoft Visual Web Developer 2010). Nevýhodou těchto prostředí je jako u Adobe komerčnost produktů. Stejně jako Adobe i firma Microsoft nabízí několika denní používání zdarma, pak je potřeba produkt koupit. Vyvíjet aplikace se dá i pomocí jiných nástrojů, ale už nenabízí tolik funkcí jako produkty od Microsoftu. Není zajištěn plnohodnotný vývoj. Vývojové nástroje běží pouze pod jakýmkoliv operačním systémem Windows. Aktuálně existuje čtyři verze, které se liší dostupností funkcí (ULTIMATE, PROFESSIONAL, PREMIUM, EXPRESS). Stejně jako u konkurence existuje verze (EXPRESS), která je

poskytovaná zdarma, ale je omezena o několik funkcí. Systémové požadavky pro bezproblémový běh prostředí na pracovní stanici jsou menší než u konkurenčního Adobe Flash Builder 4.

Nabízí tři skvělé vývojové nástroje, které nesmírně ulehčují vývoj aplikací. Bohužel se opět setkáváme s komercí. Tudíž uděluji 90 bodů.

Uživatelský pohled - Výkon aplikací je u Silverlightu mnohem nižší než u konkurenčního Adobe. Vyplývá to ze studie použitelnosti [28]. Instalace komponenty Silverlightu (zásuvný modul), abychom ji mohli plně využívat je jednoduchá. Ze stránek společnosti Microsoft si stáhneme instalační balíček, který následně naistalujeme na pracovní stanici. Pro vykreslení grafiky využívá grafických karet. Rychlost aplikace závisí na zásuvném modulu, procesoru a v neposlední řadě právě na grafické kartě. To by mohlo způsobit problém při používání zastaralých karet. Aplikace jsou spouštěny podobně jako u Adobe pomocí JavaScriptu a zásuvného modulu pro internetové prohlížeče. Nevýhodou je větší velikost aplikace. Runtime prostředí není tolik rozšířené jako u Adobe s flashem. Běží na platformách Windows a Mac OS. Pro operační prostředí Linux byl vytvořen projekt Moonlight, jehož vývoj byl svěřen firmě Novell. Toto řešení se zatím nejeví jako ideální, značně zaostává za aktuální verzí Microsoftu. Systémové požadavky jsou menší než u Adobe Flex. Silverlight lze spustit na Windows 7 (Internet Explorer 9-8, Chrome 4+, Firefox 3+), Windows Vista (Internet Explorer 9-8-7, Chrome 4+, Firefox 3+), Windows Server 2008 (Internet Explorer 9-8-7, Chrome 4+, Firefox 3+), Windows Server 2003 a XP SP2 nebo SP3 (Internet Explorer 8-7-6, Chrome 4+, Firefox 3+) a MAC OS (Firefox 3+, Safari 3+). Vzhled aplikací se dá měnit pomocí widgetů, akorát chybí přímá podpora CSS stylů. Mobilní podpora je určena především pro přístroje s operačním systémem Windows Mobile.

V tomto kritériu zaostává za konkurenčním Adobe. Hlavní nevýhodou je vazba na produkty společnosti. Dávám 85 bodů.

Technologie z pohledu vývojáře - Silverlight disponuje nejchudším SDK oproti ostatním. Zahrnuje knihovny a dokumentaci, která je možné spustit jedině s vývojovým prostředím. Flexibilita a možnosti s vývojem aplikací pomocí analyzovaného nástroje jsou obrovské a to díky tomu, že vychází z platformy .NET Framework. Silverlight nenabízí širokou multiplatformovanost a v tomto zaostává oproti konkurenci, protože je hodně vázaný na produkty Microsoftu. Základem všech aplikací Silverlight je programovací jazyk pro tvorbu uživatelského rozhraní je XAML spolu s jazyky Visual Basic a C#. V rámci

dokumentace analyzované technologie, kterou najdeme na hlavní stránce diskuzní fórum členěné do sekcí, opravdu rozsáhlou dokumentaci MSDN (Microsoft Developer Network), tutoriály a ukázkové příklady. Patří rozhodně k těm složitějším technologiím a pro její pochopení (nastudování) je potřeba určitě nejvíce času v porovnání s ostatními nástroji. Základní technologie zahrnuje kompletní seznam UI komponent, podporu kolekcí, jazyka, multimédií a další prvků. Technologie je opravdu bohatá. Podpora multimédií je u této technologie nejlepší, protože Silverlight aplikace jsou určené hlavně pro přehrávání videí a zvuků. Aplikace svým vzhledem jsou podobné Flexovým aplikacím, ale při použití nástroje pro tvorbu bohatého grafického UI se dají vytvořit aplikace, které vypadají úplně jinak než u Flexu. Rychlost vývoje je závislý na použitém jazyku pro psaní funkční části aplikace. Výhodou je možnost výběru ze dvou programovacích jazyků. Navíc Microsoft disponuje kvalitními vývojovými nástroji pro psaní Silverlight aplikací. Velikost výsledného kódu je opět závislé na výběru jazyka, ale obecně lze říci, že je průměrná. Značkovací jazyk XAML je jednoduchý na pochopení a velice přehledný.

Velkou výhodou je, že Silverlight vychází z platformy .NET Framework, nabízí možnost výběru ze dvou programovacích jazyků, kvalitní vývojové prostředí atd. Uděluji 95 bodů.

Reference - Vývojová základna je velká a to hlavně díky tomu, že Silverlight využívá technických prvků z platformy .NET Framework. Stávající vývojáři .NET mají jednodušší přechod na analyzovanou technologii. Společnost Microsoft nebude mít žádný větší problém s distribucí této technologie. Příkladem můžou být spolupráce na online přenosech ze zápasů NBA, NHL pomocí tohoto nástroje. Společnost Microsoft se chce stát jedničkou na trhu v oblasti RIA aplikací. Během dvou let bylo pomocí Silverlightu vytvořeno tisíce aplikací, které byly umístěny na nespočet webů. Televize Nova navazuje na zkušenosti získané se seriály a zpřístupňuje i živé vysílání ve formátu Silverlight. Topfun.cz - Největší česká on-line půjčovna filmů, která ve spolupráci s Centrum.cz nabízí filmy v 15 kategoriích, ke každému z produktů umísťuje krátkou upoutávku (trailer). Pro to, aby zákazník nebyl obtěžován zdlouhavým čekáním na video, rozhodli se tvůrci využít technologie Silverlight. Načítání videa je tak díky tomu v podstatě okamžité. Existuje celá řada dalších českých a zahraničních projektů využívající technologii Silverlight.

V tomto ohledu výrazně předčí konkurenci. Uděluji 100 bodů.

4.3 JavaFX

Pomocí Rich Internet Applications vypadají internetové aplikace jako desktopové programy. Společnosti Microsoft a Adobe vyvinuly technologie Silverlight a Flex. Aby i další velká dřívější společnost Sun Microsystems (dnes Oracle) nezůstala pozadu, přišla s technologií, která má název JavaFX. Oracle dodává pro JavaFX svůj vlastní programovací jazyk a dává tak k dispozici nástroje na vytváření JavaFX kódu. Ke spuštění JavaFX RIA aplikace potřebují uživatelé pouze programovací knihovnu, která spustí kód: runtime. Společnost nemá při distribuci runtime obtíže, protože Java od Oraclu je v současnosti k dispozici na 92 procentech všech počítačů. Většina uživatelů již pravděpodobně má JavaFX na svém počítači. V půlce října 2009 společnost Oracle distribuovala Java Runtime Engine 6 Update 10, který obsahuje JavaFX. Platforma Java je již navíc k dispozici i na mnoha mobilních zařízeních. Naproti tomu JavaFX pokulhává ve vývoji platformy a aplikací. S Javou je sice možné velmi dobře realizovat tradiční aplikace, při zobrazení webových obsahů mimo browser a pro multimédia však Oracle musel začít od nuly. JavaFX přináší do světa Javy nové jádrové elementy. Srdcem je skriptovací jazyk, který zobrazuje rozhraní pro aplikace a na přání je také animuje. Tři enginy implementují tyto příkazy: jeden pro webové obsahy, druhý pro renderování animací a třetí pro přehrávání filmů. U internetových komponent používá Oracle ke spouštění HTML a JavaScriptu již existující programovou knihovnu WebKit. I prohlížeč Chrome od Googlu a Adobe AIR používají WebKit, protože tato knihovna zpracovává webový kód spolehlivě a rychle. Programy v Javě mají většinou těžkopádné rozhraní, které funguje v obdélníkovém okně. Díky JavaFX se to má změnit: komponenta Scene Graph přebírá u JavaFX část Flashe plus ActionScript v AIR. Vytváří animace a vizuální efekty jako je stínování nebo změny barev. Třetí engine slouží k zobrazení multimediálních obsahů. Proto má JavaFX svůj vlastní mediální prohlížeč. Ten může pod Windows přehrávat videa pouze pomocí multimediálního rozhraní DirectShow, uživatel tedy musí mít nainstalovaný příslušný DirectShow filter pro video formát. Mezi mnohé inovace v platformě JavaFX patří funkce Drag-to-Install, která vůbec poprvé umožňuje koncovým uživatelům jednoduše přetahovat aplikace JavaFX z prohlížeče do počítače. Uživatelé budou nyní moci přesouvat aplety, běžící v prohlížeči přímo do počítače, což jim poskytne nejjednodušší možnost instalace internetových aplikací a obsahu. Vývojářské nástroje pro JavaFX poskytuje Oracle k dispozici zdarma. Patří mezi ně NetBeans jako vývojářské prostředí a Projekt Nile pro konvertování souborů Photoshopu a Illustratoru.

4.3.1 Historie

Oracle JavaFX 1.0

JavaFX 1.0 obsahuje podporu videa a tři klíčové komponenty – JavaFX Development Environment, JavaFX Production Suite a JavaFX Desktop. Přináší komplexní podporu videa napříč různými platformami a používá videokodeků společnosti On2 Technologies, Inc.

Tři klíčové komponenty [27]:

- *JavaFX Development Environment* - obsahuje kompilátor a nástroje pro runtime prostředí, knihovny pro grafiku, média, webové služby a formátovaný text spolu s integrovaným vývojovým prostředím NetBeans. Obsahuje také zásuvné moduly (plugin) JavaFX pro vývojová prostředí NetBeans i Eclipse. Dále JavaFX Development Environment obsahuje mobilní emulátor, v němž si vývojáři mohou spouštět náhled mobilních aplikací pro platformu JavaFX.
- JavaFX Production Suite je sadou nástrojů a zásuvných modulů, které umožňují webovým návrhářům výrazně zkrátit dobu produkce. Převod různých formátů médií jediným kliknutím umožňuje snadno začleňovat do aplikací JavaFX grafické objekty z návrhářských programů jako Adobe Photoshop či Adobe Illustrator. Pomocí sady JavaFX Production Suite mohou vývojáři efektivněji spolupracovat s grafickými návrháři a snadno si s nimi vyměňovat multimediální objekty. Výsledné aplikace si navíc zachovají konzistentní vzhled a charakter napříč všemi prohlížeči na různých platformách.
- *JavaFX Desktop* - využívá zásadní vlastnosti prostředí Java Platform Standard Edition, například novou implementaci javovského zásuvného modulu, která umožňuje koncovým uživatelům přetahovat spuštěné aplety z prohlížeče přímo na plochu. Poskytuje jim tak zdaleka nejjednodušší možnost instalace internetových aplikací a obsahu.

Oracle JavaFX 1.1

Další verze JavaFX 1.1 byla oficiálně vydána v únoru 2009 a jednalo se spíše o update verze předchozí. Měla odstranit velké množství chyb, které se vyskytly v první verzi. Největší novinkou dalo by se říct, byla oficiální podpora JavaFX Mobile. Umožňuje vytvářet mobilní aplikace.

Oracle JavaFX 1.2 a současnost

JavaFX SDK 1.2 je další aktualizací, tentokrát verze JavaFX SDK 1.1. Tato verze potěšila především příznivce operačních systémů Linux a OpenSolaris. Na Linux a OpenSolaris verze vyžaduje GStreamer - je framework snažící se co nejvíce ulehčit práci programátorům tvořícím multimediální aplikace. Verze přinesla následující novinky:

- Menší velikost stažení runtime JavaFX.
- Rychlejší čas spouštění aplikací JavaFX nasazených, buď jako Java Web Start aplikace nebo Java applet.
- Různé opravy chyb ke zvýšení spolehlivosti aplikací JavaFX ve webových prohlížečích Safari a Firefox na platformě Mac OS.
- Kvalitnější zpracování chyb.
- Opravy pro vizuální artefakty v aplikacích s přechody v kontrolách na některých deskách Nvidia GeForce.

V současné době je nejnovější verzí JavaFX SDK 1.3, která obsahuje:

- Nová a lepší podpora pro ovládací prvky uživatelského rozhraní a CSS
- Zvýšení výkonu v JavaFX Runtime
- Nativní písma JavaFX rodiny Amble
- Podpora rozvoje TV aplikací, včetně nové TV emulátor
- Vylepšení pro mobilní aplikace, mobilní emulátor pro Mac OS
- Podpora a zlepšení funkce pro úpravu ve vývojovém prostředí NetBeans IDE
- Nové funkce a lepší výkon v JavaFX Production Suite

4.3.2 Skriptovací jazyk - JavaFX Script

Platforma JavaFX má pro vývoj aplikací svůj specializovaný skriptovací jazyk původně pojmenovaný F3 (Form Follows Function), nyní oficiálně JavaFX Script [8]. Vývoj programů v JavaFX Scriptu ulehčuje většina používaných vývojových nástrojů (NetBeans, Eclipse). Syntaktické a další chyby jsou programátorovi předávány při překladu (tzv. compile-time errors). JavaFX Script je staticky typovaný deklarativní skriptovací jazyk, který je postaven na platformě Java. Kód JavaFX Scriptu je údajně až 5x úspornější než Javy. Syntaxe se podobá JavaScriptu. Jazyk poskytuje výhodu automatického data bindingu i plné podpory 2D grafiky, swingovských komponent a deklarativních animací. Koncovka zdrojových

souborů je „fx“. Takže pokud se jmenuje třída např. Hlavni, potom zdrojový kód napsaný v JavaFX Scriptu bude „Hlavni.fx“. Má tyto vlastnosti:

- *Staticky typovaný* – To znamená, že datový typ každé proměnné, parametru i návratové hodnoty musí být znám již při překladu.
- *Deklarativní* – Jazyk popisuje, čeho se má dosáhnout, ne však jak. Jako výsledek je jednodušší syntaxe oproti Javě. Vzhledem k tomuto rysu je JavaFX Script vhodný pro tvorbu komplexních GUI aplikací.

JavaFX Script poskytuje 4 primitivní typy: String, Boolean, Number a Integer. Proměnné jakéhokoliv typu je uvedena s klíčovým slovem var. Na rozdíl od Javy zde není třeba uvést datový typ při deklaraci, datový typ si interpret odvodí z jejího použití. Nicméně protože jde o staticky typový jazyk, nelze v průběhu životního cyklu datový typ měnit. Proměnné musí být pojmenovány v souladu s pravidly Javy. Názorná ukázka kódu JavaFX Scriptu:

```
import javafx.ext.swing.*;
SwingFrame {
    title: "Hello world";
    width: 100;
    height: 50;
    content: Label {
        text: " Hello world ";
    }
    visible: true;
}
```

Data Binding - Jedná se o obecnou techniku spojující logiku a uživatelské rozhraní. Pokud jsou správně nastaveny vazby, změni-li data hodnotu, tak elementy vázané k těmto datům se automaticky změní, platí i opačná závislost. Například změna stavu jedné GUI komponenty synchronizovaně změní jinou komponentu či datový model bez potřeby jakéhokoliv „listeneru“, který se používá třeba v Javě.

4.3.3 Vývojové prostředí – NetBeans a Eclipse

NetBeans je velice úspěšný Open Source projekt. Má rozsáhlou uživatelskou základnu, zvětšující se komunitu vývojářů a skoro 100 partnerů po celém světě. NetBeans IDE pro JavaFX poskytuje sofistikované integrované vývojové prostředí pro vytváření, prohlížení a ladění aplikací JavaFX. Editor nabízí paletu pro rychlé přidávání objektů JavaFX s transformacemi, efekty a animacemi. Také tento nástroj přichází s vlastní sadou vzorků

vytvářecích bloků a simulátorem mobilního telefonu JavaFX Mobile Emulator. Kromě vývojového prostředí je dostupná i platforma NetBeans Platform – modulární a rozšiřitelný základ pro vytváření desktopových aplikací. Dodavatelé softwaru nabízejí doplňující moduly, které lze snadno integrovat do vývojářského nástroje. Ty mohou být následně použity k vývoji jejich řešení. Poslední (nejnovější verze) vydaná verze je NetBeans IDE 7.0. Pro přehledný souhrn vlastností rozdělíme vývojové prostředí do tří částí:

- *NetBeans jako IDE* - Umožňuje správu projektů, obsahuje průzkumníka kódu, různé „typing helpery - nástroje urychlující psaní kódu“, integraci s nástroji pro čtení dokumentace typu JavaDoc, dále podporuje verzování pomocí SVN a CVS.
- *NetBeans jako vývojový nástroj pro tvorbu dynamických webových stránek* – Rozlišuje mezi lokálními a vzdálenými projekty. NetBeans při výběru místa uložení projektu také umožňuje ukládat své vnitřní soubory (soubory projektu a další soubory, které nejsou součástí výsledných stránek) do samostatného podadresáře. Tedy jinak než zdrojové kódy vyvíjené webové aplikace. To je často výhodné, neboť nechceme kód naší aplikace míchat se soubory, které k němu nepatří. NetBeans zdůrazňuje syntaxi prakticky všech webových jazyků. Vývojový program pro účely vývoje webových aplikací obsahují speciální verzi průzkumníka kódu, která zobrazuje vše potřebné. A spoustu dalších výhod.
- *NetBeans jako vývojový nástroj pro programovací jazyk Java* – Vývojové prostředí bylo primárně vyvinuto pro vývoj aplikací napsaných v jazyce Java. NetBeans dovedou plně využít potenciál Javy, jak z hlediska programového, tak i OOP modelu. Nástroje RAD lze využít jak v kontextu Java AWT, tak i Java Swing (grafické knihovny Javy). NetBeans dále plně podporují JavaDoc (dokumentace) a javovské projekty.

Eclipse je open source vývojové prostředí určené pro programování v jazyce Java. Program lze doplnit o řadu dalších rozšiřujících pluginů např. pro PHP či C++, ale i pro UML či XML. Pluginy tvoří hlavní sílu tohoto vývojového prostředí, což jej odlišuje od konkurenčních vývojových prostředí. Základní verze obsahuje standardní nástroje pro práci s Javou.

4.3.4 Runtime prostředí

Aplikace vytvořené pomocí technologie JavaFX lze spouštět přes internetový prohlížeč, z plochy počítače nebo přes mobilní zařízení. K tomu, aby aplikace správně fungovala na těchto zařízeních, musíme mít nainstalovaný virtuální stroj Javy – Java Virtual Machine (JVM). Je dostupný ve všech instalačních balíčcích Java Development Kit (JDK), Java Runtime Environment (JRE) nebo Java Standard Edition (JSE). Pro samotné spuštění JavaFX aplikace je nutné mít nainstalovaný Java Plug-in ve svém webovém prohlížeči. Pro distribuci javovských aplikací na internet slouží technologie Java Web Start (JWS). Poskytuje jednoduchou distribuci a následné spuštění aplikace jedním kliknutím přímo z webových stránek. Spuštěná aplikace přes Java Web Start, musí být uložena do jar archivu. O stáhnutí, instalaci, případnou aktualizaci a samotné spuštění se stará soubor s koncovkou „.jnlp“ ve formátu XML. Protože Java Web Start běží nezávisle na libovolném prohlížeči, je možné zavřít prohlížeč bez ukončení aplikace. Java Web Start technologie pracuje s libovolným prohlížečem a s libovolným web serverem. Každá aplikace vyvinutá může používat Java Web Start API, které umožňují aplikacím snazší přístup k souborům, k tiskárně, ke schránce atd.

4.3.5 Kritéria hodnocení

Obecné vlastnosti - Firma Sun Microsystems (dnes Oracle) vytvořila technologii, která svým způsobem měla konkurovat nástrojům od společností Adobe a Microsoft. Usnadňuje vývoj uživatelského rozhraní pro Java aplikace. Technologie JavaFX vznikla ještě později než u konkurenčního Microsoftu. Z naší nabídky se jedná o nejmladší technologii a z toho důvodu má pořád, co dohánět oproti konkurenci. V rozšíření technologie určitě nenapomohlo nedávné spojení dvou velkých firem Sun Microsystems a Oracle. Vývoj technologie se určitě nezastavil, ale probíhá mnohem pomaleji než dříve. Určitě v budoucnu se můžeme dočkat většího rozšíření a oblíbenosti technologie. Staví na základech programovacího jazyka Java a přes nevyspělost nástroje nabízí poměrně dost možností pro tvorbu robustních internetových aplikací. Architektura JavaFX obsahuje běhové prostředí v podobě appletu, prvky pro tvorbu aplikací pro různé platformy a speciální prvky pro tvorbu aplikací na mobilní zařízení, desktopové zařízení a další. Zdrojové kódy JavaFX Scriptu jsou přeloženy a zabaleny do JAR souboru. Aplikace běží jako applet, který je načten do internetového prohlížeče pomocí runtime prostředí ve formě JavaScriptu. Dodržuje klasické standardy AJAX, XML či JSON.

Velkou nevýhodou je krátká doba vývoje a JavaFX má, co dohánět oproti konkurenci. Určitě ji momentálně nepomohlo ani spojení dvou velkých společností. Píše se, že vývoj se na krátkou dobu zastavil. Dávám 70 bodů.

Vývojové prostředí - Pro vývoj JavaFX aplikací nám poslouží dvě vývojové prostředí. Jedná se o NetBeans IDE 7.0 a Eclipse. Výhodou oproti konkurenci je, že oba vývojové nástroje jsou open source, poskytují se zadarmo. NetBeans IDE obsahuje plnohodnotnou podporu technologie JavaFX. Automatické aktualizace, Builder uživatelského rozhraní, debugger a další nástroje patří k základním funkcím tohoto vývojového prostředí. Nevýhodou je dlouhá doba stáhnutí a instalace.

Existují dvě velmi kvalitní prostředí pro vývoj JavaFX aplikací: NetBeans IDE, Eclipse IDE. Vzhled a funkčnosti se vyrovnají konkurenci, ale mají velkou výhodou v tom, že jsou nabízeny zadarmo. Přiděluji 100 bodů.

Uživatelský pohled - Runtime prostředí běží všude, kde je nainstalována Java platforma. Spuštění JavaFX aplikací je relativně snadné a při prvním spuštění je potřeba stáhnout a nainstalovat malý objem dat. Od použití této technologie může odradit vývojáře zdlouhavá instalace v případě, že nemají na své pracovní stanici nainstalovanou Java platformu. Od toho se odvíjí i rychlost aplikace. Velkou výhodou je uplatnění JavaFX u mobilních zařízení (JavaFX Mobile). V tomto ohledu jednoznačně předčí konkurenci. Aplikace jsou spuštěny uvnitř internetového prohlížeče pomocí zásuvného modelu. Spuštění aplikací je relativně pomalejší než oproti ostatním aplikacím. Rozšířenost JavaFX těží z rozšíření runtime prostředí Javy. Systémové požadavky jsou nenáročné a asi nejnížší mezi technologiemi. Běží na platformách Windows 7 (Internet Explorer 6+, Firefox 3+), Windows Vista SP2 (Internet Explorer 6+, Firefox 3+), Windows XP SP3 (Internet Explorer 6+, Firefox 3+), Mac OS (Firefox 3+, Safari 3+), Linux (Firefox 3+), Solaris (Firefox 3+). Vzhledově aplikace vychází z Java knihovny Swing a jsou rozšířeny o vektorovou grafiku či animaci.

Ani v tomto bodě nějak nezaostává za konkurencí, dokonce předčí i Silverlight v tom, že je více multiplatformní. Dávám 90 bodů.

Technologie z pohledu vývojáře - Velkou výhodou je možnost oddělení programování vývojáře od grafika díky grafickým formátům. S multiplatformovostí nemá JavaFX sebemenší potíže. Díky platformě Java, která je nezávislá na operačním systému, běží JavaFX aplikace všude. SDK obsahuje jako ostatní technologie základní sadu knihoven, kompilátor,

navíc demo aplikace a tutoriály. Skriptovací Jazyk JavaFX Script vychází z jazyka Javy a proto vývojářům nedělá sebemenší problémy jej pochopit a naučit se v něm vyvíjet aplikace. Při tvorbě aplikací lze využívat i samostatný jazyk Java, což je velká výhoda. Velikost výsledného kódu je menší v porovnání s Adobe Flex. Na oficiálních stránkách lze nalézt API dokumentace, tutoriály, rozsáhlé diskuzní fórum a pár demo aplikací. Knihovny kromě základních prvků obsahují třídy pro zpracování dat, asynchronní komunikaci, tvorbu grafiky (geometrické tvary, 2D grafické prezentace, stíny atd.) a syntaxi jazyka. Nastudování nástroje je o něco málo rychlejší než u konkurenčního Silverlightu či Flexu. S podporou multimédií u této technologie je to podobné jako u Adobe Flex (Video formáty VFL a WMV, Audio formáty MP3 a VMA). Pomocí technologie JavaFX je mnohem snazší vyvíjet graficky animované aplikace než internetové eshopy. Další nevýhodou je nepřehlednost jazyka JavaFX Script, který může dělat problémy začínajícím programátorům.

Velkou výhodou je vazba na oblíbený programovací jazyk Java a velmi kvalitních vývojářských nástrojů. Uděluji 95 bodů.

Reference - U obou konkurentů (Microsoft, Adobe) sem zmínil pár projektů, aplikací, které byly vytvořeny pomocí analyzovaných technologií. V tomto ohledu má společnost Oracle s nástrojem JavaFX co dohánět. Neexistuje tolik velkých projektů vytvořených pomocí této technologie. Díky popularitě programovacího jazyka Java, ze kterého vychází skriptovací jazyk JavaFX Script, je vývojářská základna rozsáhlá. Srovnat se dá s konkurencí u Microsoftu. Stejně jako u konkurence není vůbec žádný problém sehnat informace ohledně dané technologie.

Krátká doba vývoje zapříčinila, že prozatím neexistuje významných projektů vycházejících z této technologie. Určitě má Oracle, co dohánět oproti konkurenci. Dávám 85 bodů.

4.4 Google Web Toolkit

Společnost Google je známá tím, že se snaží proniknout snad do všech oblastí IT, ve kterých „se něco děje“. Dalo se tedy čekat, že vyvine platformu pro vývoj webových aplikací a nabídne ji veřejnosti k použití jako otevřenou. Google Web Toolkit (GWT) je unikátní řešení „All-In-One“ ve smíchání všeho potřebného k tvorbě Rich Internet Applications na platformě Java. Myšlenka GWT je jednoduchá. Poskytnout framework, ve kterém si uživatel může jednoduše vytvořit webovou aplikaci složenou ze znovupoužitelných UI komponent

(renderovaných jako HTML) a logiky (ve formě JavaScriptu) na straně webového prohlížeče a asynchronního volání serverové logiky skrze AJAX. [3] To vše na platformě Java s využitím jazyka, knihoven a vývojových nástrojů této platformy. GWT je vhodné využít pro tvorbu webových aplikací desktopového typu, není ale velký problém v něm vyvíjet i rozsáhlé dynamické webové stránky. Systém Google Web Toolkit se skládá z několika stavebních kamenů:

- Vývojový shell
- asynchronní implementace vzdáleného volání
- emulace Java platformy
- kompilátor Javy do JavaScriptu
- komponenty pro tvorbu uživatelského rozhraní

Hlavní komponentou, která plní klíčovou roli je bezesporu kompilátor. Umožňuje přeložit pro internetový prohlížeč zdrojový kód v programovacím jazyku Java do JavaScriptu. UI komponenty se rozdělují do dvou kategorií: widgets a panels. Zatímco mezi widgets patří tlačítka, seznamy, tabulky, formuláře, vstupní pole a další, tak panels představují komponenty pro rozložení layoutu aplikace (StackPanel, HorizontalPanel, TabPanel, DockPanel a další). Aplikace vytvořené v GWT mohou běžet ve dvou módech: *hosted* a *web*. Aplikace ve webovém módu pracuje stejně jako klasická. V druhém módu běží jako klasická java aplikace ve speciálně upraveném GWT prohlížeči. Aby aplikace mohla komunikovat se serverem (volat aplikační logiku), musí implementovat nějakou formu vzdáleného volání metod – Remote Procedure Call. V prostředí internetového prohlížeče to lze udělat posíláním XML přes HTTP, a to nejlépe asynchronní formou (požadavek nevyžaduje znovunačtení okna prohlížeče) pomocí klientského objektu XMLHttpRequest. Kromě výše uvedených základních komponent nabízí Google Web Toolkit i další užitečné funkce, které usnadňují tvorbu Rich Internet Applications. Mezi ně patří internacionalizace, command line nástroje, JUnit integrace, JSNI, skládání UI a stylování, management historie atd. [23]

4.4.1 Historie

Google Web Toolkit byl vyvinut roku 2006 pod licencí Apache Software License. V roce 2008 byla uvolněna další verze 1.5, která podporovala pouze Javu 1.4 na straně klienta. S touto verzí přišla podpora anotací, for-each cyklu, generika a další novinky. Další verzi, která se objevila na trhu, byla verze 2.0. Umožňovala tzv. hosting mode, který uměl spouštět aplikace přímo v internetovém prohlížeči s nainstalovaným plug-inem. Používal se

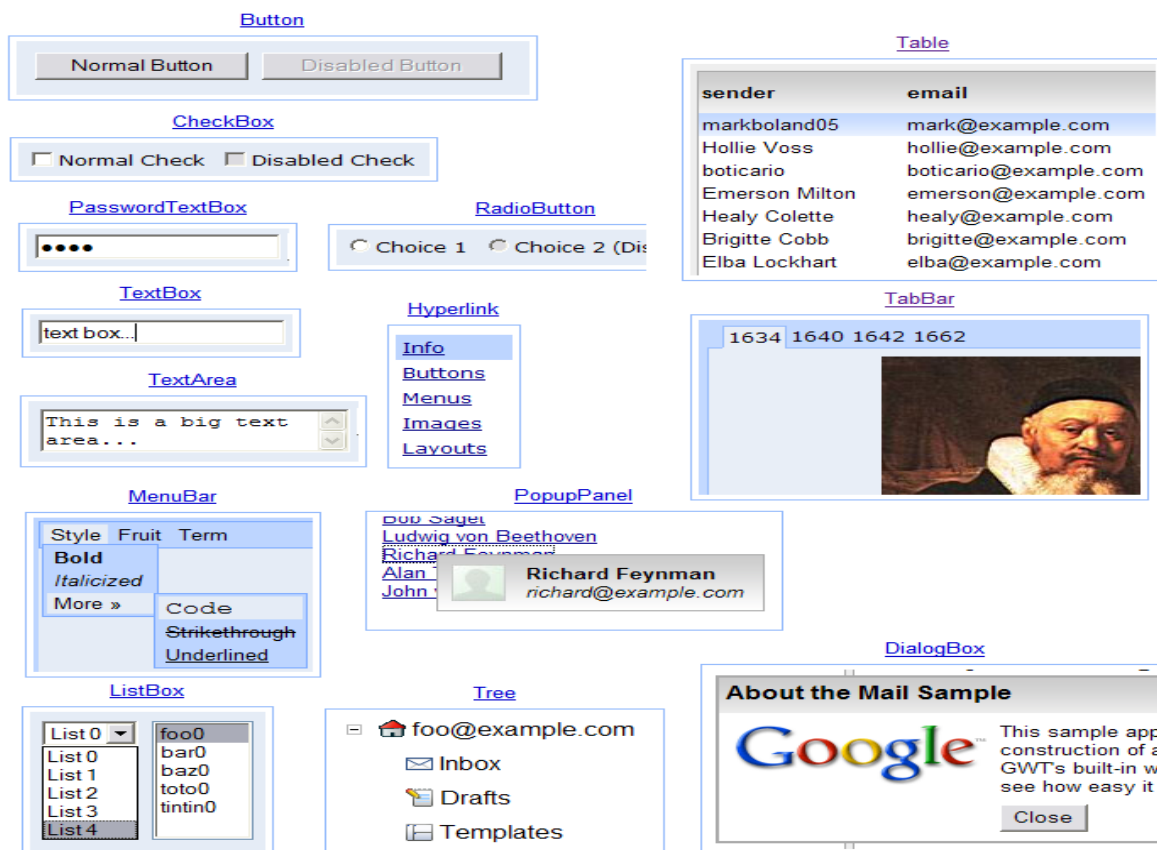
speciální prohlížeč založený na jádrech Internet Explorer a Mozilla/Gecko. Tento mód navíc ladí aplikaci ve fázi vývoje. Klientská část není přeložena do JavaScriptu, ale „byte-kód“ Javy je interpretován pomocí Java Virtual Machine a nic nám nebrání v debuggování. Mezi tím, co byla uvolněna verze 2.1, se neodehráli ve vývoji zásadní novinky. Tato verze podporovala architekturu (návrhový vzor) Model - View - Presenter, o kterém jsem se zmínil v předešlé kapitole. Vydání nejaktuálnější verze 2.2 obsahuje integrovaný drag and drop UI designer (součást Google Plugin pro Eclipse). Google Web Toolkit 2.2 podporuje HTML5 specifické funkce jako jsou canvas prvky, které umožňují dynamické vykreslování 2D tvarů, bitmapové obrázky ve vymezené oblasti webové stránky a "video", "audio" tagy.

4.4.2 Programovací jazyk Java

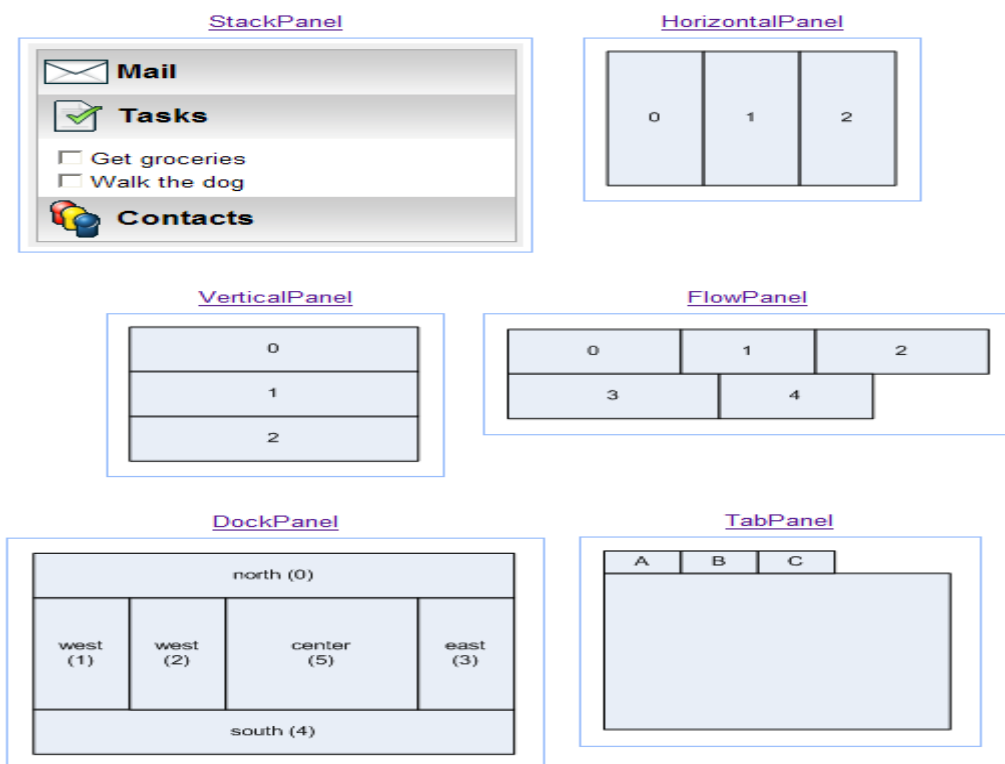
Zdrojové kódy aplikace Google Web Toolkit jsou napsané v Javě. Java je mladý programovací jazyk, který nachází zajímavá uplatnění především na Internetu. Je programovací jazyk nezávislý na platformě. Vyvíjí ho společnost SUN Microsystems (dnes Oracle) a je zdarma dostupný pro různé operační systémy (Windows, Linux, Solaris). Nezávislost na operačním systému a hardware je zajištěna způsobem kompilace. Zdrojové kódy nejsou překládány do strojového kódu procesoru, ale přepracovány do tzv. byte-kódu. Java Virtual Machine provede převedení předzpracovaného byte-kódu na strojový kód daného procesoru při spuštění Java programu. Odměnou za kompilaci je řádově vyšší výkonnost (rychlost). Oproti JavaScriptu, který se výhradně používá pouze na webu, naproti tomu je programovací jazyk Java univerzální používaný mimo jiné i na webu. Java je plně objektově orientovaný programovací jazyk. Proměnné mají přesně určený datový typ, tedy hodnoty, kterých mohou nabývat. Jazyk je podobný jazyku C++. Základní rozdíl je v tom, že Java je plně objektová, neexistují v ní globální proměnné, vše musí být součástí nějakého objektu (nebo třídy). V jazyce Java jsme odstíněni od hardware, takže např. nemáme přímou kontrolu nad pamětí a není potřeba uvolňovat alokovanou paměť.

4.4.3 Komponenty pro tvorbu uživatelského rozhraní

Jak sem se již zmínil výše, tak UI komponenty se dají rozdělit do dvou hlavních skupin - widgets a panels. Widgets je skupina základních UI komponent, jako jsou tlačítka, tabulky, vstupní políčka, rozbalovací seznamy a podobně. Oproti tomu ve skupině panels se nalézají komponenty pro definici layoutu aplikace, tedy rozvržení komponent na stránce. Jednoduché ukázky widgetů a panelů v GWT:



Obr. 4.4.1 – Ukázka komponent z *Widgets* [23]



Obr. 4.4.2 – Ukázka komponent z rodiny *Panels* [23]

4.4.4 Vývojové prostředí

Google společně s Google Web Toolkit poskytuje podporu vývoje aplikací přímo ve vývojovém prostředí Eclipse. Pro lepší podporu vytváření GWT aplikací je potřeba si nainstalovat Google Plugin pro Eclipse. Jedná se o sadu nástrojů pro vývoj softwaru, který umožňuje Java vývojářům rychle navrhovat, vytvářet, optimalizovat a nasadit tzv. cloud-based aplikací. Plugin v současné době podporuje Google App Engine rozvoj a je kompatibilní s Eclipse 3.6. Pro práci s GWT doporučuji použít některý ze silných vývojových nástrojů pro tvorbu v Javě. Existuje podpora pro IntelliJ IDEA, Eclipse i NetBeans.

4.4.5 Kritéria hodnocení

Obecné vlastnosti - Architekturu tvoří dvě části: JRE knihovny pro GWT a komponenty pro uživatelské rozhraní GWT. Zdrojové kódy Javy jsou přeloženy kompilátorem do JavaScriptu. Google Web Toolkit se vyvíjí delší dobu oproti své konkurenci. Ani společnost Google není pozadu a stále vyvíjí nové verze a aktualizuje stávající. Budoucnost technologie není ohrožena. Podporuje standardy CSS, HTML, JSON, AJAX, XML, SOAP a další.

GWT vychází z programovacího jazyka Java a předčí konkurenční JavaFX v dlouhodobějším vývoji. Přiděluji 90 bodů.

Vývojové prostředí - GWT aplikace lze vyvíjet ve vývojovém nástroji Eclipse IDE. Výhodou je, že vývojové prostředí se poskytuje zdarma. Vývojář má možnost pomocí jednoho příkazu vygenerovat strukturu aplikace a projekt přímo určený pro Eclipse. Ten můžeme naimportovat do vývojového nástroje a využít všechny dostupné funkce vývojového nástroje Eclipse. Stáhnutí a instalace prostředí trvá mnohem méně než u NetBeans IDE. Eclipse umožňuje zvýrazňování a kontrolu syntaxe, kvalitní systém nápověd a řešení problémů, propojení s hosted módem včetně krokového ladění.

Skvělý vývojový nástroj Eclipse spolu se zásuvným modulem dělá vynikajícího pomocníka pro tvorbu GWT aplikací. Navíc je vše zdarma. Dávám 95 bodů.

Uživatelský pohled - Systémové požadavky jsou nízké, běží na platformách Windows 7, Windows Vista, Windows XP, Linux, Mac OS a na internetových prohlížečích typu Internet Explorer, Firefox, Safari a Opera. Aplikace vypadají vzhledově podobně jako desktopové aplikace vytvořené v programovacím jazyce Java, díky použití Java knihovny Swing. Nevýhodou je i malý počet základních vizuálních komponent. Google o podpoře Google Web Toolkitu na mobilní platformy vůbec neuvažuje.

V tomto ohledu má hodně nedostatků, i když aplikace běží na nejznámějších platformách. Dávám 80 bodů.

Technologie z pohledu vývojáře - Jedná se o open source pod Apache License. Jednou z velkých výhod této aplikace je její jednoduchost. Naučit se ji dá určitě rychleji než třeba technologii Adobe Flex či Silverlight. Aplikace GWT se vyvíjí v populárním jazyce Java. Pomocí překladače je bytekódu přeložen do JavaScriptu. Jednotlivé komponenty uživatelského prostředí se programují v Javě a syntaxe je podobná jako u vytváření softwarových Java produktů v knihovně Swing. Vývojové nástroje obsahují stejně jako u Adobe Flex knihovny a kompilátor. Google Web Toolkit obsahuje vlastní jednoduché běhové prostředí podporující logování. Rychlost aplikace je závislá na interpretaci JavaScriptu na klientském počítači. Dokumentace zahrnuje popis hlavních knihoven generovaných z JavaDoc. Je dostupná i online dokumentace přímo na stránkách Googlu. Nechybí ani diskusní fórum, FAQ, tutoriály, ukázkové příklady aplikací nebo kompletní přehled komponent uživatelského rozhraní. Oproti Flexu obsahuje méně uživatelským komponent a to díky omezení převodu komponent uživatelského rozhraní z Javy do JavaScriptu. S podporou multimédií tato technologie dosti zaostává za konkurencí. Pomocí kvalitního vývojového prostředí a Javy je dosaženo rychlého vývoje aplikací. Problém může vzniknout při oddělení složitého vzhledu od funkční logiky. Multiplatformovanost je vysoká, podporuje všechny významné.

Hlavní výhodou je jednoduchost, kvalitní vývojové prostředí, dobře známý jazyk Java. Špatná podpora multimédií a mobilů. Uděluji 90 bodů.

Reference - Společnost Google je na internetu pojem. Hlavně díky svému internetovému vyhledávači a určitě i v oblasti RIA aplikací patří mezi přední firmy na trhu. Mezi nejznámější aplikace využívající GWT jsou Google AdWords a Orkut. Aplikace není třeba představovat, protože lidem zajímavější se o vývoj webových aplikací jsou velice pověštné. Na stránkách Googlu lze najít dostatek informací ohledně Google Web Toolkitu. Společnost nebude mít problém s distribucí technologie.

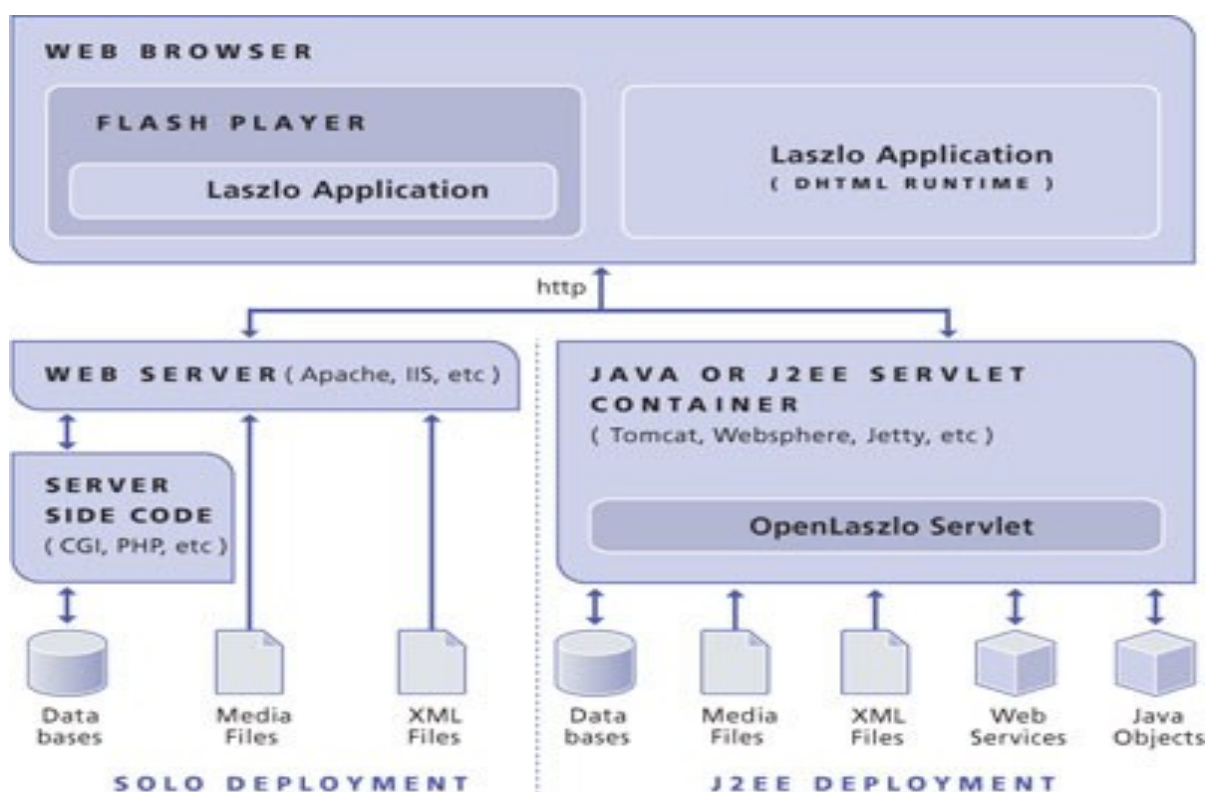
Silná společnost Google, mnoho známých aplikací vytvořených pomocí GWT. Přiděluji 90 bodů.

4.5 OpenLaszlo

Rich Internet Applications jsou spojovány především s technologiemi Flex, Silverlight a JavaFX. Existují však i další nástroje a jedním z nich je open source platforma OpenLaszlo od firmy Laszlo Systems. Předešlé uvedené technologie vyvíjely velké společnosti, které jsou známé široké veřejnosti. Na Internetu najdeme mnoho internetových stránek, které se zajímají o uvedené nástroje. Na rozdíl od těchto řešení je technologie OpenLaszlo a firma Laszlo Systems méně populární. Z tohoto důvodu nenalezneme mnoho tuzemských vývojářů a českých webových stránek, kteří by se zabývali touto technologií. I když u nás se ještě tolik neuchytila, ve světě je běžně využívána. Technologie umožňuje vytvořit snadno uživatelské rozhraní pomocí předdefinovaných komponent a výsledek vyexportovat jak do podoby SWF (Flash), tak i do dynamického HTML nebo Javy. OpenLaszlo by tak mohlo hrát velkou roli při vývoji aplikací, u nichž je potřeba, aby fungovala i na zařízeních, která Flash nepodporují. Pyšní se filosofií „write once, run anywhere“ [2], což znamená podporu pro všechny významné systémy a webové prohlížeče. SDK OpenLaszlo se skládá z Java servletu, JavaScriptové knihovny (LFC-Laszlo Foundation Class) a kompilátoru [24].

- *Server* - je Java Servlet zajišťující podporu pro SOAP5, XML-RPC6 a JavaRPC7 požadavky. Dále je schopen např. překódovat různé formáty tak, aby byly spustitelné prostřednictvím Flash Playeru.
- *Kompilátor* - je napsaný v programovacím jazyce Java se stará o překlad zdrojových textů z LZX jazyka do binárních souborů spustitelných v cílovém prostředí.
- *Knihovna LFC* - zahrnuje komponenty uživatelského rozhraní, data binding a síťové služby. Podporuje také tvorbu animací a layoutu.

Programování probíhá pomocí programovacího jazyka LZX. Kombinuje XML a JavaScript. Přesto, že se tato platforma vyvíjí už poměrně dlouho, chybí mocné a komplexní integrované vývojové prostředí, existuje pouze starší rozšíření pro Eclipse, na kterém se již ale dlouhou dobu nepracuje. Aplikace, které vytvoříme prostřednictvím technologie OpenLaszlo mohou být dostupné jako klasické Adobe Flash nebo DHTML aplikace. Označují se jako „SOLO applications“. [2]



Obr. 4.5 – Architektura OpenLaszlo [24]

4.5.1 Historie

Na internetových stránkách firmy Laszlo Systems najdeme dokumentaci pro tři verze. Verze Open Laszlo 3.4.0 přidávala dvě nové funkce: Jednak se jednalo o podporu CSS2 a streamování medií – streamování Flash videa (flv) přes http, obousměrný audio-video streaming s Flash Media Serverem nebo Red 5, podpora kamery a mikrofonu. Tato verze se také dočkala vylepšení dokumentace. Další plně kvalifikované vydání se nazývá OpenLaszlo 4.2.0.3, která je doporučena pro všechny platformy pro vývoj aplikací v SW8, SW9 a DHTML. Vydání opravuje více než 30 chyb. Aktuální verze 4.9.0 implementuje infrastrukturu pro podporu nové komponenty. Poskytuje nové funkce pro vytváření a zavádění mobilních aplikací. Verze 4.9.0 podporuje moderní prohlížeče Safari, Opera, Chrome, Internet Explorer a běží na platformách Windows XP, Windows 7, Linux a Mac OS.

4.5.2 Značkovací jazyk LZX

Pro technologii OpenLaszlo byl vyvinutý speciální značkovací jazyk LZX – JavaScript Description Language, který je kompilován pro různé runtime prostředí. Značkovací jazyk vychází ze syntaxe XML a je hodně podobný jazyku XUL14. Pro vzájemné působení UI prvků s uživatelem a aplikační logiky na straně klienta je použitý

JavaScript. LZX API podporuje animační efekty, obsluhování událostí, data binding, obsahuje také velké množství předdefinovaných komponent a widgetů.

4.5.3 Vývojové prostředí

Jak GWT aplikace, tak OpenLaszlo aplikace lze vytvářet v libovolném textovém editoru. Nejběžnějším vývojovým prostředím pro technologii OpenLaszlo je Eclipse IDE s pluginem IDE4Laszlo. Aplikace se dají vytvářet i pomocí jednoduché pluginů do vývojového prostředí NetBeans.

4.5.4 Hodnocení kritérií

Obecné vlastnosti - Možnosti technologie jsou sice velké díky tomu, že se dokáží spustit v mnoha runtimeových prostředích, ale nedokáže využít všech jejích vlastností. Vzhledem k tomu, že první verze byla vyvinuta v roce 2002, jedná se už o poměrně zavedenou technologii. Architektura zahrnuje runtime knihovny přeložené do všech aplikací a skýtají služby běhového prostředí. Navíc obsahuje prezentační vrstvu, která se stará o vykreslování komponent uživatelského rozhraní. Spuštění aplikací je rozdílné oproti předešlým technologiím. Uživatel se dotazuje přímo zdrojový kód LZX souboru. Další mechanismus zajistí server, který najde dotazovaný soubor spolu s ostatními prvky (obrázky, knihovny atd.), zkompiluje ho a spustí uživateli. Podporuje standardy XML, JSON, SOAP, CSS, AJAX, ale nepodporuje internacionalizaci.

Technologie se vyvíjí dlouhou dobu, hlavní výhoda je běh ve více runtime prostředích. Uděluji 90 bodů.

Vývojové prostředí - S vývojovým prostředím pro tvorbu aplikací pomocí technologie OpenLaszlo je dosti špatné. Existuje pouze jeden plnohodnotný nástroj Eclipse IDE se zásuvným modulem IDE4Laszlo, který usnadňuje tvorbu OpenLaszlo aplikací. Bohužel na vývoji tohoto rozšíření se přestalo pracovat, takže nám nezbývá nic jiného než použít jakýkoliv textový editor. Stejně jako u GWT je možné psát OpenLaszlo aplikace v libovolném textovém editoru.

Bohužel zatím neexistuje plnohodnotné vývojové prostředí pro tvorbu OpenLaszlo aplikací. Dávám 60.

Uživatelský pohled - OpenLaszlo aplikace lze spouštět v nejrůznějších runtime prostředí a díky podporované technologii DHTML i v nejpoužívanějších webových prohlížečích. I když

technologie nemá problém se spuštěním aplikací, tak nedokáže využít všechny možnosti technologií např. Flash Playeru. Kombinace Flash Playeru a DHTML usnadňuje instalaci aplikací. Systémové požadavky jsou průměrné, běží na platformách Windows 7, Windows XP, Linux, Mac OS a ve webových prohlížečích Internet Explorer, Firefox, Opera, Safari, Chrome. Bohužel technologie zatím nefunguje v aktuálně nejnovějších verzích jako je IE 9 nebo Firefox 4. Možnou výhodou technologie může být jejich vzhled, který se ubírá jiným směrem než ostatní. Mohou stylem připomínat Linuxové aplikace.

Hlavní výhodou je jiný směr vzhledu než ostatní, běh ve více běhových prostředích a různých webových prohlížečích, bohužel zatím není podpora těch nejaktuálnějších prohlížečů. Přiděluji 80 bodů.

Technologie z pohledu vývojáře - Hlavní výhodou technologie OpenLaszlo je bezesporu multiplatformovanost. Usnadňuje tvorbu RIA aplikací nezávislé na platformě. Programátor OpenLaszlo aplikací je může nabídnout v různých alternativách, aniž by použil různé nástroje a postupy. Další výhodou dané technologie je rozsáhlá knihovna komponent a podpora animací, které lze snadno zakomponovat do aplikací. Aplikace OpenLaszlo jsou napsány jazykem LZX, který je postavený na jazyku XML a je překompilován do různých runtime prostředí. Značkovací jazyk LZX je intuitivní, přehledný a nedisponuje žádnou zákeřnou syntaxí. SDK nezaostává za ostatními, jak je tomu třeba u Silverlightu. Tradičně zahrnuje základní sadu knihoven jednotlivých komponent, příklady, tutoriály a dokumentaci. Dokumentace na oficiálních stránkách technologie OpenLaszlo je vytvořena formou statistických stránek. Obsahuje programátorskou příručku, instalační příručku, demo aplikace, diskuzní fórum a přehled komponent UI. Pochopení technologie OpenLaszlo je určitě rychlejší než např. u JavaFX. Ani tato technologie nezůstává pozadu s bohatým seznamem knihoven pro zpracování událostí, grafiky, animace, získávání dat ze serveru, multimédií atd. S podporou multimédií je to stejné jako u Flexu a JavaFX. Technologie OpenLaszlo hlavně vsází na jednoduchou tvorbu a rychlost vývoje, který výrazně brzdí kvalitní vývojové prostředí. Velikost výsledného kódu je mnohem menší než u jiných technologiích. Rychlost je ovlivněna Adobe Flash Playerem. Na podpoře pro mobilní zařízení se usilovně pracuje.

Vývoj technologie značně potřebuje kvalitní vývojové prostředí, jinak se jedná o zajímavé technické řešení s mnoha možnostmi. Uděluji 80 bodů.

Reference - S porovnáním s ostatními technologiemi je dostupnost informací nejobtížnější. Je to dáno tím, že za nástrojem OpenLaszlo nestojí významná firma jako u ostatních. Navíc se

zdá, že u českých vývojářů se technologie netěší velké oblíbenosti. Na internetu jsem nenašel české stránky, které by se zajímali o toto technické řešení. Ve světě je tato technologie daleko populárnější. Odhaduji, že vývojářská základna bude velmi malá, díky částečné nedostupnosti informací o daném nástroji. Informace o technologii lze hlavně z dokumentace, která je zpracovaná opravdu výborně. Mezi nejznámější aplikace vytvořené pomocí technologie OpenLaszlo jsou Pandora, Wal-Mart a FNAC.

V tomto bodě bylo jasné, že firma Laszlo Systems nemůže konkurovat velkým společnostem, ale např. existuje mnohem více známých projektů než je tomu u JavaFX. Přiděluji 80 bodů.

4.6 Matice prostých a vážených užitečností

Veškeré dosažené výsledky jsou přehledně zobrazeny v následujících dvou maticích (prosté a vážené užitečností). Slovní okomentování výsledků je provedeno v závěru diplomové práce.

Varianta A - Adobe Flex

Varianta B - MS Silverlight

Varianta C - JavaFX

Varianta D - Google Web Toolkit

Varianta E - OpenLaszlo

Č.	Kritérium	Varianty					
		A	B	C	D	E	M
1.	Obecné vlastnosti	100	95	70	90	90	100
2.	Vývojové prostředí	85	90	100	95	60	100
3.	Uživatelský přístup	95	85	90	80	80	100
4.	Techn. z pohledu vývojáře	95	95	90	90	80	100
5.	Reference	95	100	85	90	80	100

Tab. 4.6.1 - Matice prostých užitečností

M - Nejvyšší možná dosažená hodnota

Č.	Kritérium	Váha	Varianty					
			A	B	C	D	E	M
1.	Obecné vlastnosti	3	300	285	210	270	270	300
2.	Vývojové prostředí	4	340	360	400	380	240	400
3.	Uživatelský přístup	3	285	255	270	240	240	300
4.	Techn. z pohledu vývojáře	5	475	475	450	450	240	500
5.	Reference	3	285	300	255	270	240	300
	Celková užitečnost		1685	1675	1585	1610	1230	1800
U	Relativní užitečnost		93,61%	93,06%	88,06%	89,44%	68,33%	100%

Tab. 4.6.2 - Matice vážených užitečností

M - Nejvyšší možná dosažená hodnota

U- Relativní užitečnost (%)

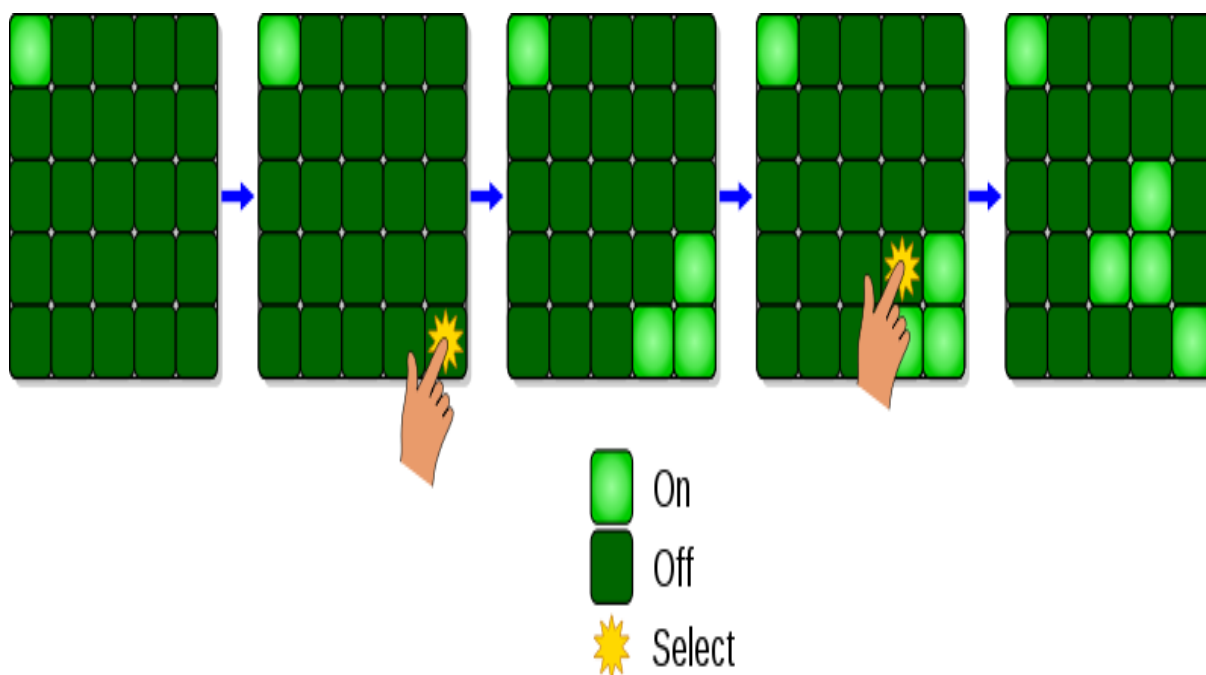
5. Návrh a realizace demonstrační aplikace

Předmětem této části diplomové práce je návrh a realizace demonstrační aplikace pomocí vybrané technologie pro tvorbu robustních internetových aplikací. Smyslem kapitoly je ukázat praktický příklad RIA aplikace. K vytvoření použiji JavaFX technologii. Proč jsem si zvolil právě tuto technologii? Těch důvodů je hned několik. I když se v analýze nástroj JavaFX neumístil na předních místech, jedná se o opravdu kvalitní technologii, která má svůj potenciál. Hlavní příčinou špatného umístění ve srovnávání je jednak mladistvost a nevyzrálost technologie a dále málo známých aplikací vytvořených právě touto technologií. Hlavní příčinou výběru JavaFX je kvalitní vývojové prostředí (NetBeans IDE). S tímto nástrojem mám nemalé zkušenosti a navíc je poskytováno zdarma. Dalším důvodem jsou dostatečné zkušenosti s programovacím jazykem Java, ze kterého vychází skriptovací jazyk JavaFX Script. A v neposlední řadě mohu poukázat na to, že pokud bychom srovnávali jednotlivé technologie podle kritérií, jenž jsou nejdůležitější v případě začínajícího programátora, který si chce vytvořit první aplikaci pomocí RIA nástrojů, tak technologie JavaFX zaujme jedno z předních umístění. Možnou nevýhodou je nutnost nainstalovaného Java Virtual Machine na osobním počítači ke spuštění aplikace. Na závěr bych chtěl ještě

zdůraznit, že se jedná o demonstrační aplikaci, nikoliv o aplikaci nasazenou do ostrého provozu. Cílem není vytvořit špičkový program, který budou hojně využívat uživatelé internetu nebo firmy, ale pouze na jednoduchém příkladu demonstrovat možnosti a vlastnosti RIA aplikací.

5.1 Návrh aplikace - „Nenech bílou vyhrát“

Pro praktickou ukázkou jsem si vymyslel jednoduchou chytlavou inteligentní hru, která má zvláštní název: „Nenech bílou vyhrát“. Smyslem hry je odstranění všech bílých políček z hracího pole o velikosti 7x7 čtverečků. Na začátku každé hry se vygeneruje náhodné hrací pole, které se skládá z bílých a žlutých čtverečků. Bílých se musíme postupně zbavit tak, aby na hracím poli zůstaly pouze čtverečky se žlutou barvou. Po stisknutí bílého políčka se změní barva na žlutou a naopak. Celý princip se zdá být velice jednoduchý, ale opak je pravdou. Bylo by velice jednoduché, kdyby uživatel jen klikal na bílá políčka a ty se mu měnily před očima ve žluté kostky. Aby uživatel zapojil mozkové buňky, situace je stížená tím, že po kliknutí na bílé políčko (i žluté) se změní barva i přilehlým čtverečkům. Přilehlé znamená nacházející se ve směru hlavních světových stran (sever, jih, východ, západ) od stlačeného čtverečku. Úplně stejný princip funguje i v opačném případě, kdy hráč stiskne žlutě vybarvené políčko. S tím rozdílem, že nezůstane žluté, ale změní se na bílé. Pro úplné pochopení funkcionality aplikace nám poslouží následující obrázek (obr. 5.1):



Obr. 5.1 - Princip funkcionality hry „Nenech bílou vyhrát“ [29]

Pokud se hráč dokliká k situaci, že mu na hracím poli nezbyde žádné políčko s bílou barvou (pouze žluté), tak vyhrál a může se pustit do další úrovně. S každou přibývajícím úrovní stoupá náročnost ve smyslu rostoucího počtu bílých políček. Hra nemá konečnou úroveň, tudíž ji uživatel může hrát donekonečna, resp. dokud jej nepřestane bavit. V několika odstavcích jsem tedy popsal funkcionalitu hry, kterou jsem si navrhl. Dalšími požadavky návrhu aplikace jsou, aby měla intuitivní, přehledné, rychlé ovládání a pěknou úvodní obrazovku. Hra by se měla spouštět jako applet v HTML stránce anebo jako samostatná aplikace pomocí technologie Java Web Start.

5.2 Realizace aplikace - „Nenech bílou vyhrát“

Aplikace je vytvořena ve vývojovém prostředí NetBeans IDE (Obr. 5.2) a je přiložena k diplomové práci v podobě ZIP archívu. Projekt se skládá ze čtyř složek a dvou souborů:

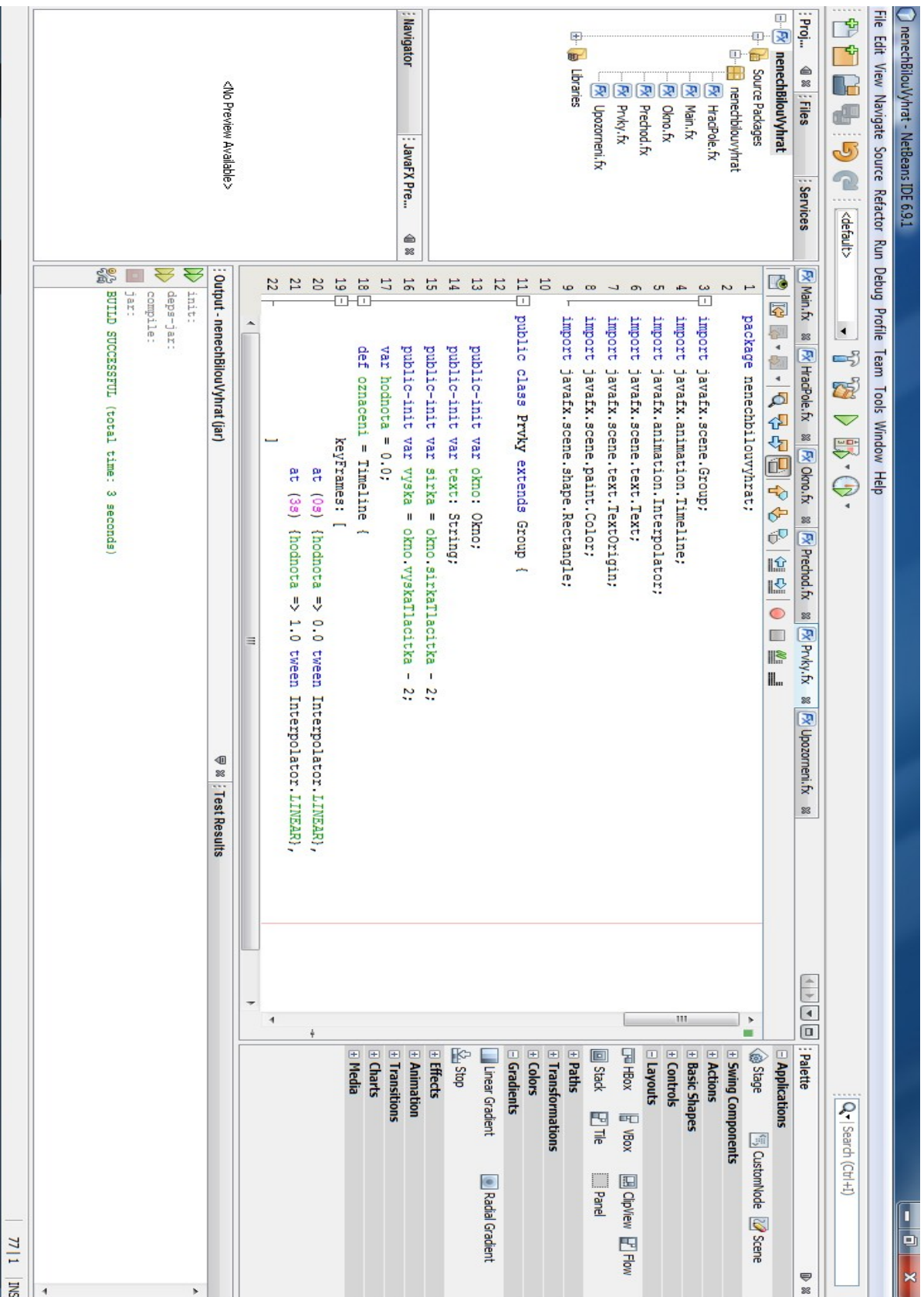
- Adresář *build* - obsahuje veškeré přeložené zdrojové soubory (*.class), kopírují se sem konfigurační soubory (tento adresář může být smazán).
- Adresář *nbproject* - zahrnuje informace, které jsou důležité pro vývojové prostředí NetBeans, aby mohlo otevřít danou aplikaci jako projekt
- Adresář *dist* - jsou zde veškeré soubory pro distribuci aplikace. Najdeme zde HTML soubor, ve které je výsledná aplikace spuštěna jako applet. K tomu, aby mohla aplikace běžet jako applet, musí se v adresáři nacházet ještě jeden soubor „*_browser.jnlp“. Dále zde najdeme další soubor s koncovkou „.jnlp“, který umožňuje spustit aplikaci pomocí technologie Java Web Start a samotnou aplikaci v souboru s koncovkou „.jar“.
- Adresář *src* - obsahuje veškeré zdrojové soubory aplikace
- Soubor *build.xml* - výchozí konfigurační soubor pro Ant
- Soubor manifest - je speciální soubor, který může obsahovat informace o souborech zabalených do JAR souboru.

Jak jsem uvedl výše, tak v adresáři *src* se nachází veškeré zdrojové soubory aplikace. Hra se skládá z šesti souborů „.fx“.

- *HraciPole.fx* - jak již z názvu vyplývá, jedná se o soubor, který obsahuje třídu reprezentující hrací pole a veškerou funkcionalitu s ním. Ve třídě nalezneme metody, které zajišťují vygenerování hracího pole, nastavení barev jednotlivým políčkům, kontrolu a vymazání pole a spoustu dalších metod.

- *Main.fx* - Hlavní soubor, který obsahuje metody ke spuštění a zavření aplikace. Navíc zjišťuje, zda je hra spuštěna uvnitř internetového prohlížeče či nikoliv.
- *Okno.fx* - soubor, ve které je třída reprezentující layout aplikace. Nastavuje hodnoty rozměrů a umístění jednotlivých grafických prvků. Seřizuje velikost písma jednotlivým komponentám atd.
- *Prechod.fx* - obsahuje třídu reprezentující animaci, jenž se provede na hlavní obrazovce. V rámci třídy je metoda, která provádí celou animaci. Nastavuje vlastnosti pohybujících se grafických prvků.
- *Prvky.fx* - soubor v němž je ukryta třída, která reprezentuje ostatní grafické prvky, zejména jednotlivé tlačítka a nastavuje jim vlastnosti, jako jsou např.: událost při kliknutí, zvýraznění tlačítka při najetí myši, jednotlivé barvy atd.
- *Upozorneni.fx* - najdeme zde třídu, která reprezentuje hlášení při úspěšném zvládnutí úrovně. Nastavuje vlastnosti jednotlivým textovým komponentám, vykresluje obsah, nastavuje rozmístění atd.

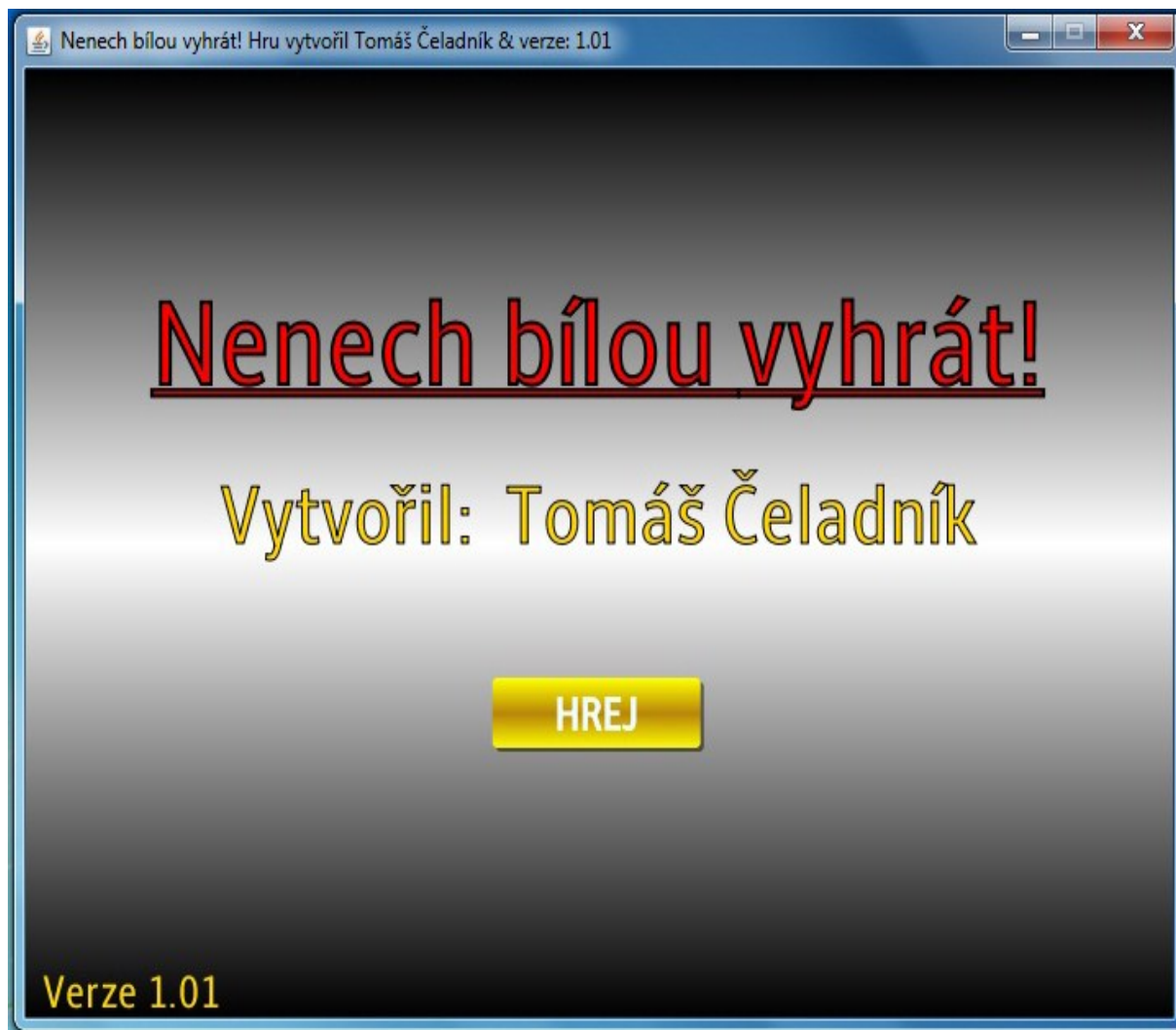
Ještě než ukončím tuto podkapitolu je potřeba zmínit, že aplikace by měla běžet i v mobilních zařízeních, které mají nainstalovanou Javu. Bohužel, ale aplikace obsahuje např. různé druhy barev, které nedokáže zařízení správně zobrazit a tím pádem výsledná aplikace nevypadá hezky po vizuální stránce. S funkcionalitou by neměl být žádný problém. V rámci kapitoly zde nebudu uvádět žádné zdrojové kódy nebo detailně popisovat jednotlivé třídy, protože celá aplikace je přiložena k práci a každý má možnost nahlédnout do těchto adresářů či souborů.



Obr. 5.2 - Realizace aplikace v NetBeans IDE 6.9.1

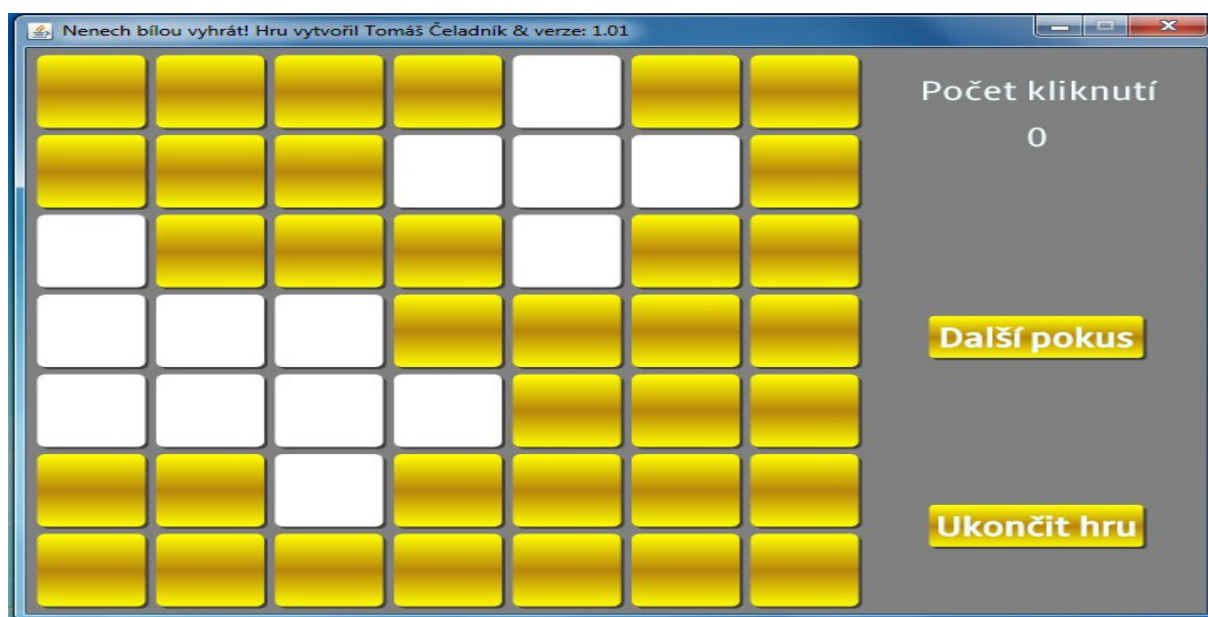
5.3 Výsledná podoba aplikace

Při každém spuštění hry se na začátku objeví úvodní obrazovka (Obr. 5.3.1) s animací, která se skládá z textových popisků (název hry, autora aplikace, verze) a tlačítka „HREJ“, které po stisknutí spustí hru.



Obr. 5.3.1 - Úvodní obrazovka hry „Nenech bílou vyhrát“

Po spuštění hry se před námi zobrazí (Obr. 5.3.2) hrací pole o rozměrech 7x7 čtverečků, kde jsou vidět vygenerované barvy jednotlivým políčkům. Dále zde nalezneme dvě tlačítka: „Další pokus“ a „Ukončit hru“. V případě, že si hráč neví rady, jak odstranit všechny bílé čtverce z hracího plátna, má možnost kliknout na prvně jmenované tlačítko, které zajistí nově vygenerované hrací pole. Druhé jmenované slouží k ukončení hry a zavření okna. Ještě nesmím zapomenout na textovou komponentu „Počet kliknutí“, která zobrazuje aktuální počet kliknutí.



Obr. 5.3.2 - Hrací pole hry „Nenech bílou vyhrát“

V případě úspěšného zvládnutí (všechny žluté čtverečky) se nám zobrazí průhledná komponenta s hlášením (Obr. 5.3.3), stupněm úrovně a button „Další úroveň“, který po stisknutí vygeneruje větší počet bílých políček v závislosti na hodnotě úrovně.



Obr. 5.3.3 - Výpis úspěšného zvládnutí

V několika málo krocích jsem popsal celou funkcionalitu, návrh a výsledné soubory realizace aplikace „Nenech bílou vyhrát“.

6. Závěr

Nejen mladší, ale i starší generace tráví stále více času na internetu vyhledáváním informací, komunikací s přáteli, obchodováním, hraním her a dalšími možnostmi, které současný internet nabízí. Ať si to člověk připouští nebo ne, internet se stává součástí našeho volného času. Proto společnosti neustále vymýšlejí a vyvíjejí nové technologie, které usnadňují a zefektivňují možnosti poskytování služeb uživatelům. Rozhodně mezi tyto nástroje pro usnadnění patří RIA technologie, které umožňují tvorbu robustních (bohatých) internetových aplikací.

Tato diplomová práce se zabývá zmapováním současných technologií pro tvorbu RIA aplikací a vymezení jejich oblastí působnosti. V teoretické části se zabývám vývojem a součástmi aplikací v prostředí internetu. Dále jsem vysvětlil význam zkratky RIA, uvedl významné vlastnosti a popsal výhody, nevýhody RIA aplikací.

V praktické části jsem uvedl samotnou analýzu technologií pro tvorbu robustních (bohatých) internetových aplikací. Na základě mého uvážení jsem si vybral v současnosti pět nejpoužívanějších technologií. Čtyři z těchto nástrojů vyvíjejí opravdu významné společnosti v oblasti informačních technologií: Adobe - Adobe Flex, Microsoft - Silverlight, Oracle - JavaFX a Google, Google Web Toolkit, Laszlo Systems - OpenLaszlo. Každé z těchto technologií jsem věnoval samostatnou část, ve které jsem popsal jejich vlastnosti, historii, aktuální verze, vývojové prostředí a plno dalších zajímavostí. Dále jsem použil vícekritériální analýzu k jejich srovnání.

Pořadí srovnávaných technologií podle vícekritériální analýzy:

1. Adobe Flex - 93,61%
2. MS Silverlight - 93,06%
3. Google Web Toolkit - 89,44%
4. Oracle JavaFX - 88,06%
5. OpenLaszlo - 68,33%

Pomyslným vítězem méj analýzy se stala technologie Adobe Flex. Hlavní předností této technologie je runtime prostředí Adobe Flash Player, které je multiplatformní a je rozšířené na všech moderních operačních systémech. Další výhodou technologie je dlouhodobější vývoj oproti ostatním firmám, které ji musí neustále dohánět. Jazyk ActionScript je jednoduchý a dá se snadno naučit. Existuje celá řada významných projektů

vytvořených právě touto technologií. Určitě stojí za zmínku i kvalitní vývojové prostředí, které nabízí řadu efektivních nástrojů pro tvorbu RIA aplikací, bohužel za plnohodnotný nástroj si je potřeba připlatit.

Zajímavostí je, že těsně za ní skončil nástroj od společnosti Microsoft. I když je technologie Silverlight mnohem mladší než Flex, během své krátké historie se podílela tvorbě mnoha významných projektů. Hlavně v oblasti streamování videa nemá konkurenci. Společnost nabízí opravdu nepřeberné množství kvalitních vývojových nástrojů, které předčí konkurenci v mnoha ohledech (design, programovací nástroje), ale za kvalitu se platí. Další výhodou je možnost použití více programovacích jazyků (VB.NET, C#). Velkou nevýhodou Silverlightu je vazba na produkty Microsoftu a s tím souvisí menší rozšířenost runtimeového prostředí Silverlightu. I když je Adobe pořád jedničkou v oblasti RIA, Silverlight ji rychle dohání. Bronzovou medaili získala technologie Google Web Toolkit. Hlavní výhodou je jednoduchost vývoje, který vychází ze spojení oblíbenosti programovacího jazyka Java a kvalitního vývojového prostředí Eclipse IDE, které lze rozšířit o zásuvný modul a tím získat plnohodnotné vývojové prostředí pro tvorbu GWT aplikací. Navíc je poskytováno zdarma.

Velkým překvapením je umístění technologie JavaFX až na čtvrtém místě. Bohužel tato technologie je nejmladší a nevyzrálá ze všech srovnávaných a s tím souvisí i nízký počet aplikací vytvořených pomocí daného nástroje. Dle mého názoru zbrzdilo vývoj i nedávné sloučení dvou významných společností. Předpokládám, že se v budoucnu dočkáme mnohem většího rozmachu této technologie a to díky mobilní podpoře. V této oblasti dominuje před konkurencí. Na posledním místě se umístila technologie OpenLaszlo. Už před analýzou jsem očekával, že se tento nástroj neumístí na předních místech díky velké konkurenci velkých společností. I když se svým způsobem jedná o zajímavé technické řešení, které těží ze spolupráce Adobe Flash Playeru a možností běhu na více runtime prostředích, tak neexistence plnohodnotného vývojového nástroje pro tvorbu OpenLaszlo aplikací ji řadí na poslední místo v méj analýze. Další nevýhodou této technologie je malá základna vývojářů a špatná dostupnost informací na internetu o daném nástroji.

Cílem diplomové práce bylo zmapování a srovnání současných technologií pro tvorbu robustních internetových aplikací, vymezení oblasti působnosti RIA nástrojů a v poslední části návrh a realizace demonstrační aplikace. Všechny tyto cíle práce byly naplněny. Tuto diplomovou práci lze v budoucnu rozšířit o porovnání s dalšími technologiemi pro vývoj RIA aplikací a zkomponování technických specifikací jednotlivých technologií. Doufám, že

výsledek mé práce bude prospěšný všem, kteří si chtějí prohloubit své znalosti v oblasti RIA technologií.

Seznam použité literatury

Knihy

1. BALDERSON, J.; ENT, P.; HEIDER, J.; PREKASKI, T.; SUGDEN, T.; TRICE, A.; HASSOUN, D.; BERKOVITZ, J.; *Professional Adobe Flex 3*. Wrox Press, 2009. 1488 s. ISBN 978-0-470-22364-2
2. COREMANS, Ch. *AJAX and Flash Development with OpenLaszlo*. BrainySoftware, 2006. 350 s. ISBN 0-9752128-6-9
3. DEWSBURY, R. *Google Web Toolkit Applications*. Prentice Hall, 2007. 574 s. ISBN 978-0-321-50196-7
4. KEYES, J. *X Internet, The Executable And Extendable Internet*. Auerbach Publications, 2007. 372 s. ISBN 0-8493-0418-0
5. LUBOSLAV, L. *Silverlight : výukový průvodce tvorbou interaktivních aplikací*. Brno : Computer Press, 2010. 464 s. ISBN 978-80-251-2716-2
6. MACDONALD, M.; SZPUSZTA, M. *ASP.NET 3.5 a C# 2008. Přel.* RNDr. Pokorný, J.; Gregor, J. Zoner Press, 2008. 1584 s. ISBN 978-80-7413-008-3
7. TRETOLA, R.; BARBER, S.; ERICKSON, R. *Professional Adobe Flex 2*. John Wiley and Sons, 2007. 687 s. ISBN 0470102675.
8. WEAVER L., J. *JavaFX Script: Dynamic Java Scripting for Rich Internet/Client-side Applications*. 1st ed. Apress, 2007. 200 s. ISBN 1-59059-945-4
9. ZAKAS, N.; McPeak, J.; Fawcett, J. *Professional Ajax*. Wiley Publishing, Inc., 2006. 406 s. ISBN 978-0-471-77778-6

Elektronické publikace

10. SYMBIO: Internetová agentura. Rich Internet Application. [online]. c1999-2010 [cit. 2011-04-01]. Dostupný z WWW: <<http://www.symbio.cz/slovník/rich-internet-application.html>>
11. CLINICK, A. Remote Scripting, Microsoft Corporation. [online]. 1999, [cit. 2011-03-19]. Dostupný z WWW: <<http://msdn.microsoft.com/en-us/library/ms974566.aspx>>
12. PICHLÍK, R. *Rich Internet Application* [online]. 2005, [cit. 2011-02-13]. Dostupný z WWW: <<http://interval.cz/clanky/rich-internet-application/>>

13. BERNARD, B. *Rich Internet Applications v roce 2008*. [online]. 2008, [cit. 2011-03-19]. Dostupný z WWW: <<http://interval.cz/clanky/rich-internet-applications-v-roce-2008/>>
14. ČERMÁK, M. *Na velikosti záleží*. [online]. 2008, [cit. 2011-02-13]. Dostupný z WWW: <<http://www.lupa.cz/clanky/na-velikosti-zalezi/>>
15. KODÝTEK, P. *HISTORIE INTERNETU*. [online]. 2006, [cit. 2011-02-03]. Dostupný z WWW: <<http://www.webdesign.paysoft.cz/clanky/2006/historie-internetu/>>
16. BERNARD, B. *Úvod do architektury MVC*. [online]. 2009, [cit. 2011-02-03]. Dostupný z WWW: <<http://zdrojak.root.cz/clanky/uvod-do-architektury-mvc/>>
17. ADOBE SYSTEMS: Vývojářské centrum Flex. *What's new in Flex 4*. [online]. 2010, [cit. 2011-04-01]. Dostupný z WWW: <http://www.adobe.com/devnet/flex/articles/flex4sdk_whatsnew.html/>
18. VŠB-TU Ostrava, Microsoft © 2010. *Lekce 1 - Technologie a RIA*. [online]. 2010, [cit. 2011-04-01]. Dostupný z WWW: <<http://silverlight.cs.vsb.cz/01-technologie-a-ria.aspx>>
19. BERNARD, B. *Adobe Flex - co je a není*. [online]. 2008, [cit. 2011-04-07]. Dostupný z WWW: <<http://interval.cz/clanky/adobe-flex-co-je-a-co-neni/>>
20. Grossman G., Huang E. *ActionScript 3.0 overview*. [online]. 2006, [cit. 2011-03-02]. Dostupný z WWW: <http://www.adobe.com/devnet/actionscript/articles/actionscript3_overview.html>
21. ADOBE SYSTEMS: Flash Platform. *Adobe Flash Player*. [online]. 2011, [cit. 2011-04-07]. Dostupný z WWW: <<http://www.adobe.com/cz/products/flashplayer/>>
22. VŠB-TU OSTRAVA, Microsoft © 2010. *Lekce 2 - Vývojové prostředí*. [online]. 2010, [cit. 2011-04-07]. Dostupný z WWW: <<http://silverlight.cs.vsb.cz/02-vyvojove-prostredi.aspx>>
23. PICHLÍK, R. *Google Web Toolkit*. [online]. 2009, [cit. 2011-04-14]. Dostupný z WWW: <<http://interval.cz/clanky/google-web-toolkit/>>
24. LASZLO SYSTEMS: společnost. *Architecture*. [online]. 2006-2009, [cit. 2011-04-14]. Dostupný z WWW: <<http://www.openlaszlo.org/architecture>>
25. HERCEG, T.; JECHA, T. *Visual Basic. NET FAQ: Co to je Visual Basic?* [online]. 2007, [cit. 2011-04-14]. Dostupný z WWW: <<http://www.vbnet.cz/faq.aspx>>

26. REDAKCE: INTERVAL.CZ Silverlight - Co je Silverlight? [online]. 2008, [cit. 2011-04-14]. Dostupný z WWW: <<http://interval.cz/clanky/silverlight-co-je-silverlight/>>
27. VIRKLIS, a.s. The Wings Agency. *Sun Microsystems uvádí platformu JavaFX 1.0 s bohatými multimediálními funkcemi pro 800 milionů počítačů s Javou na trhu.* [online]. 2009, [cit. 2011-04-14]. Dostupný z WWW: <<http://www.virklis.eu/cs/press/javafx1.html>>
28. MALLOIRE, J. *Flash Vs Silverlight: A Usability Evaluation*, Bournemouth University. [online]. 2008, [cit. 2011-04-14]. Dostupný z WWW: <<http://www.jamesmallorie.com/Flash%20Vs%20Silverlight.pdf>>
29. WIKIPEDIE: The Free Encyclopedia. *Lights Out (game)*. [online]. 2004, [cit. 2011-04-20]. Dostupný z WWW: http://en.wikipedia.org/wiki/Lights_Out_%28game%29
30. LE HÉGARET, P. *Document Object Model (DOM)*. [online]. 2005, [cit. 2011-04-11]. Dostupný z WWW: <<http://www.w3.org/DOM/>>
31. KOSEK ml., J. *CGI-skripty*. [online]. 1999, [cit. 2011-03-26]. Dostupný z WWW: <<http://www.kosek.cz/clanky/iweb/06.html>>
32. KOTALA, Z.; TOMAN, P. *Applet*. [online]. 2001, [cit. 2011-03-26]. Dostupný z WWW: <<http://v1.dione.zcu.cz/java/sbornik/17.html>>
33. JANOVSKEÝ, D. *Úvod do JavaScriptu*. [online]. 2011, [cit. 2011-04-10]. Dostupný z WWW: <<http://www.jakpsatweb.cz/javascript/javascript-uvod.html>>
34. CAJTHAML, M. *Silverlight nebo Flash?* [online]. 2007, [cit. 2011-04-15]. Dostupný z WWW: <<http://www.symbio.cz/clanky/silverlight-nebo-flash.html>>
35. ADOBE SYSTEMS: *Výhody platformy Flash*. [online]. 2011, [cit. 2011-04-15]. Dostupný z WWW: <<http://www.adobe.com/cz/flashplatform/benefits/>>

Seznam zkratek

RIA – Rich Internet Applications

AJAX - Asynchronous JavaScript and XML

HTML - HyperText Markup Language

ARPA - Advanced Research Projects Agency

CERN - Conseil Européen pour la recherche nucléaire

WWW - World Wide Web

LCS - Laboratory of Computer Science

MIT - Massachusetts Institute of Technology

INRIA - Institut National de Recherche en Informatique et en Automatique

URI - Uniform Resource Identifier

URL - Uniform Resource Locator

HTTP - HyperText Transfer Protocol

FTP - File Transfer Protocol

CGI - Common Gateway Interface

JVM - Java Virtual Machine

XML - eXtensible Markup Language

DOM - Document Object Model

W3C - World Wide Web Consortium

ECMA - European Computer Manufacturers Association

JSP - JavaServer Pages

ASP - Active Server Pages

PHP - Hypertext Preprocessor

API - Application Programming Interface

MVC - Model View Controller

JSON - JavaScript Object Notation

MPV - Model View Presenter

CSS - Cascading Style Sheets

DHTML - Dynamic HyperText Markup Language

OS - Operating System

XHTML - eXtensible HyperText Markup Language

TCP/IP - Transmission Control Protocol - Internet Protocol

P2P - Peer-To-Peer

GUI - Graphical User Interface

XUL - XML User interface Language

SDK - Software Development Kit

IDE - Integrated Development Environment

RTF - Rich Text Format

SWF - ShockWave Flash

MXML - Macromedia eXtensible Markup Language

FXG - Flash XML Graphics

HD - High-Definition

AIR - Adobe Integrated Runtime

SOAP - Simple Object Access Protocol

FLV - Flash Video

GWT - Google Web Toolkit

UI - User Interface

GIS - Geographic Information System

WPF/E - Windows Presentation Foundation/everywhere

XAML - eXtensible Application Markup Language

WMV - Windows Media Video

WMA - Windows Media Audio

WCF - Windows Communication Foundation

OOB - Out Of Browser

SEO - Search Engine Optimization

GPS - Global Positioning System

COM - Compact Object Model

GPU - Graphics Processing Unit

DRM - Digital Rights Management
PDA - Personal Digital Assistant
PC - Personal Computer
VBA - Visual Basic for Application
CIL - Common Intermediate Language
JIT - Just It Time
WYSIWYG - What You See Is What You Get
LINQ - Language INtergrated Query
MS SQL - MicroSoft Structured Query Language
CLR - Common Language Runtime
MSDN - MicroSoft Developer Network
NBA - National Basketball Association
NHL - National Hockey League
F3 - Form Follows Function
SVN - SubVerSion
CVS - Concurrent Version System
OOP - Object Oriented Programming
RAD - Rapid Application Development
AWT - Abstract Widows Toolkit
UML - Unified Modeling Language
JDK - Java Development Kit
JRE - Java Runtime Environment
JSE - Java Standard Edition
JWS - Java Web Standard
JAR - Java ARchive
IT - Information Technology
JSNI - JavaScript Native Interface
FAQ - Frequently Asked Questions
LFC - Laszlo Foundation Class

Prohlášení o využití výsledků diplomové práce

Prohlašuji, že

- jsem byl seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou práci užít (§ 35 odst. 3);
- souhlasím s tím, že diplomová práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že bibliografické údaje o diplomové práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, diplomovou práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne

.....
jméno a příjmení studenta

Adresa trvalého pobytu studenta:

.....

Přílohy

Příloha č. 1

CD

- tento dokument ve formátu DOCX a PDF
- vytvořenou aplikaci „Nenech bílou vyhrát“ v podobě ZIP archívu
- vlastní obrázky použité v dokumentu
- obrázky tabulek v dokumentu